

RESUMEN

Este proyecto de tesis se concibió partiendo de la necesidad de introducir al mundo de la tecnología de la información a una institución de enseñanza de lenguajes, el Lincoln College of Foreign Languages para la cual el autor de este proyecto trabajó por algunos años como profesor de inglés. La institución prestó su auspicio al proyecto por la relación de confianza y amistad construida a través de los años con sus Directivos.

Existían procesos críticos para los cuales se manejaban métodos tradicionales e inadecuados como archivos físicos y hojas Excel que presentaban un riesgo muy alto para la integridad de estos datos. En repetidas ocasiones debido a cortes de energía o virus informáticos, estas hojas Excel se vieron afectadas con la consiguiente pérdida de datos y malestar de los clientes.

El proyecto se enfocó en la concepción de un sistema informático que maneje los siguientes parámetros:

- Almacenamiento seguro de datos
- Recuperación eficiente y rápida de datos
- Automatización de los procesos más críticos de sus operaciones como institución educativa.
- Manejo de una intranet corporativa.
- Manejo de perfiles de acceso y usuarios.
- Manejo de interfaces tipo RIA (Rich Internet Applications).
- Uso de tecnologías open source y/o free software

Se realizó el levantamiento de requerimientos en una etapa inicial a través de entrevistas a los usuarios potenciales del producto de software, información que sirvió para armar los elementos de la metodología de desarrollo y establecer de una mejor manera las necesidades del negocio. La experiencia obtenida a través de los años también sirvió como una ayuda a este esfuerzo.

Debido a que el objetivo del proyecto se basaba principalmente en la utilización de tecnologías open source para las aplicaciones tipo RIA (Rich Internet Applications), se procedió a investigar varias tecnologías, y se concluyó definitivamente que se utilizaría el framework Extjs por su robustez, funcionalidad, excelentes resultados y principalmente porque mediante el uso de adaptadores permitía encapsular a cualquier otra tecnología. Extjs tiene como prerequisite un conocimiento mediano a avanzado de javascript y css además de una curva de aprendizaje alta con largas horas de investigación y aplicación (trial and error). En ocasiones era necesario ponerse en contacto con el equipo de desarrollo del core y con programadores de todo el mundo para compartir experiencias, resolver diferentes situaciones y colaborar.

El proyecto se encuentra concluido y ha sido un desafío.

CAPITULO I

INTRODUCCIÓN

1.1. TEMA

Desarrollo de una intranet corporativa para la automatización de procesos mediante el uso de tecnologías open source php y ajax. Caso de estudio: Lincoln College of Foreign Languages.

1.2. JUSTIFICACIÓN

El Lincoln College of Foreign Languages es una institución dedicada a la enseñanza de idiomas entre ellos el Inglés, Francés, Italiano, Portugués, etc.

Esta institución imparte clases en sus instalaciones pero al mismo tiempo mantiene convenios con varios colegios y empresas de la capital como parte de un proceso de outsourcing de la enseñanza de idiomas, es decir, estas instituciones educativas confían las operaciones de su respectiva área de idioma extranjero a un operador como Lincoln College of Foreign Languages, mientras tanto que las empresas contratan clases para sus empleados en la propia empresa o en las instalaciones del instituto.

Pero con el advenimiento de estos convenios, también se ha creado la necesidad de modernizar y automatizar procesos que actualmente se realiza con la ayuda de la herramienta Microsoft Excel.

En diversas ocasiones, debido a daños en los computadores se ha perdido información que luego debió ser recuperada recurriendo a grandes archivos, lo cual significó pérdida de tiempo y productividad.

Dado lo anterior, es un riesgo continuar con las mismas políticas y procedimientos cuando se habla de la información de los clientes de la institución.

Otro aspecto que se ha tomado en cuenta es el alto costo de las licencias de software para productos como Oracle, Sql Server, Sybase, Borland, entre otros; cuyo desempeño es muy similar a las aplicaciones open source que se van a utilizar en este proyecto.

El sistema propuesto podría permitir a Lincoln College mantener su base de clientes y posiblemente expandir sus operaciones. Además de un incremento en competitividad que le permitirá un sostenido desarrollo.

1.3. OBJETIVO GENERAL

Desarrollar una intranet corporativa para la automatización de procesos críticos del Instituto de Idiomas Lincoln College of Foreign Languages para que este alcance un mejor nivel de operatividad y competitividad mediante el uso de tecnologías open source.

1.4. OBJETIVOS ESPECÍFICOS

- Utilizar una metodología efectiva para la resolución del problema.
- Desarrollar un sistema con tecnologías open source ajax php
- Elaborar la respectiva documentación de usuario y técnica.
- Realizar las pruebas en ambiente de desarrollo.

1.5. ALCANCE

El proyecto busca brindar una solución informática al instituto de idiomas Lincoln College of Foreign Languages mediante la aplicación de la ingeniería de software y el uso de tecnologías de código abierto también denominadas open source con el uso de ajax.

Las áreas de operación del sistema serán:

- Ingreso y mantenimiento de datos
- Matriculación
- Ingreso y mantenimiento de calificaciones
- Gestión de cursos
- Sueldos de profesores y asistencia de alumnos
- Generación de reportes

Cabe recalcar que no se ha llegado a ningún acuerdo para la implantación de este sistema en dicha institución, ni compensación alguna, más se trata de un apoyo como requisito de un proyecto de tesis.

CAPITULO II

MARCO TEÓRICO

2.1 El valor del software

Cuadro 2.1 : El Software según Grady Booch

“El software es el combustible del cual los negocios modernos se basan para diferentes propósitos cada uno de ellos en su campo de acción respectiva. El software brinda una gran asistencia en la creación, acceso y visualización de la información en maneras diversas antes inconcebibles. Globalmente, el progreso del software ha ayudado al crecimiento de la economía mundial. En una escala más humana, los productos de software han curado a los enfermos y dado la voz a los mudos, movilidad para los inválidos y oportunidad a los menos capaces. Desde todas estas perspectivas el software es una parte indispensable de nuestro mundo moderno”

^[1] Grady Booch, "Leaving Kansas," IEEE Software 15(1), Jan.–Feb. 1998, pp. 32–35.

Las buenas noticias para los profesionales del software es que las economías globales dependen cada vez más en este y las demandas de hoy hacen que estos se expandan en medida, complejidad, distribución e importancia, lo cual no es un problema por la tecnología al momento.

Las malas noticias es que la expansión de estos sistemas en medida, complejidad, distribución e importancia empuja los límites de lo que nosotros en la industria del software conocemos como desarrollo.

Añadiendo al problema está el hecho que los negocios continúan demandando más productividad y calidad con desarrollo más rápido y despliegue. Además, la cantidad de personal calificado de desarrollo no está a la par con la demanda. El resultado de esto es que construir y mantener software es duro y la situación se está agravando. La construcción de software de calidad en un proceso predictivo y repetitivo es más difícil aún.

2.2 La Ingeniería de Requerimientos

La Ingeniería de requerimientos se preocupa de identificar, modelar, comunicar y documentar los requerimientos de un sistema, y los contextos en los cuales el sistema será utilizado. Los requerimientos describen lo que se va a hacer pero no cómo éstos van a ser implementados. Existen muchas técnicas disponibles para asegurar que los requerimientos sean **completos, consistentes y relevantes**.

El objetivo de la Ingeniería de Requerimientos es ayudar a saber qué construir antes que empiece el desarrollo del sistema para así prevenir refacciones costosas. Este objetivo se basa en dos principios:

1. Mientras más tarde se descubra un error, más costoso será corregirlo.
2. Es posible determinar un set estable de requerimientos antes que el diseño del sistema y la implementación comiencen.

2.2.1 Proceso de la Ingeniería de Requerimientos

Este consiste de cinco actividades principales

- Obtención
- Análisis y Negociación
- Documentación

- Validación
- Evolución de requerimientos

2.2.2 Obtención de requerimientos

Se refiere a la identificación de los requerimientos y el alcance del sistema a través de consultas a usuarios y partes interesadas. El alcance del sistema afecta grandemente a este proceso. Una completa comprensión de las diferentes áreas como son: dominio de la aplicación, necesidades del negocio, restricciones, el problema en sí mismo; es esencial para divisar lo que se va a desarrollar.

Las técnicas más importantes para la obtención de requerimientos son:

2.2.3 La Entrevista

La entrevista es un método para descubrir hechos y opiniones de usuarios potenciales y otras partes interesadas del sistema bajo desarrollo. Los errores y malentendidos pueden ser identificados y clarificados de esta manera. Hay dos clases de entrevistas.

- **La entrevista cerrada**, cuando el o los desarrolladores tienen un set de preguntas previamente elaboradas para encontrar respuestas específicas.
- **La entrevista abierta**, cuando no se tiene las preguntas previamente elaboradas y los desarrolladores y clientes o partes interesadas discuten abiertamente acerca de la manera cómo quieren que el sistema trabaje.

No existe un orden establecido para aplicar ambos tipos de entrevistas, usualmente se empieza con algunas preguntas que se necesita discutir y estas llevan a otras nuevas preguntas. La

ventaja de la entrevista es que ayuda al desarrollador a obtener amplia información. La desventaja es que el análisis de la información relevante puede ser difícil y puede haber conflictos en la información brindada por los entrevistados.

2.2.4 Lluvia de ideas

La lluvia de ideas es una posibilidad de desarrollar soluciones creativas relacionadas a un cierto tópico. Normalmente, la lluvia de ideas es una actividad de grupo pero también se lo puede realizar individualmente. El proceso de la lluvia de ideas contiene dos fases:

- La fase de generación, donde las ideas son recolectadas, no discutidas ni criticadas.
- La fase de evaluación, donde las ideas son discutidas.

Las ideas deben ser desarrolladas rápidamente y deben ser amplias sin importar cuán lógicas parezcan al momento ya que este pensamiento puede desembocar en nuevas ideas. La técnica de lluvia de ideas lleva a un mejor entendimiento del problema para todos y un sentido de participación colaborativa y exitosa una vez concretado.

2.2.5 Observación y análisis social

Los métodos observacionales envuelven a un investigador que observa a los usuarios realizando su trabajo en un campo de estudio en particular y tomando notas sobre lo observado. La observación puede ser directa, donde el investigador está presente durante la tarea, o indirecta, donde la tarea es observada utilizando otros métodos como por ejemplo una

grabadora de video. Este método es muy útil en las fases tempranas de recolección de requerimientos para obtener datos en función de la calidad y no de cantidad.

La observación permite ver lo que los usuarios realmente hacen dentro del contexto de estudio. La observación directa tiene como ventaja el permitir enfocarse en áreas de interés específico. En cambio la observación indirecta tiene como ventaja la detección de aspectos que no fueron detectados con la observación directa.

2.2.6 Análisis y Negociación de Requerimientos

El análisis de requerimientos está orientado a detectar una necesidad (la necesidad para realizar el requerimiento), consistencia (los requerimientos no deben ser contradictorios), totalidad (ningún servicio o restricción debe ser ignorado) y factibilidad (en términos del contexto del proyecto, presupuesto y cronogramas asignados al desarrollo del proyecto).

Los conflictos en los requerimientos son resueltos a través de priorización. Los requerimientos que parecen problemáticos son analizados por los desarrolladores y los clientes para llegar a soluciones reales que pueden incluir hasta cambios a otros requerimientos. Las técnicas principales para el análisis de requerimientos son Sesiones JAD, Priorización y Modelamiento.

2.2.7 Priorización de requerimientos

En un proyecto con tiempos no muy amplios, recursos limitados y altas expectativas del cliente es esencial realizar las características más valiosas lo más temprano posible. Cuando la fecha de terminación del proyecto se aproxima pero no todas las características están implementadas, algunas de éstas deben ser desechadas. Por eso es importante definir que características son más relevantes al comienzo. La priorización de requerimientos debe ser

realizada por el cliente únicamente. El desarrollador elabora con el cliente una lista de características y el cliente se encarga de seleccionar cuales son las más importantes que definitivamente deben incluirse en el desarrollo y traen máximo beneficio a su empresa o negocio. Una vez seleccionadas, los desarrolladores añaden su punto de vista en relación a riesgos técnicos, costos o dificultades para que el cliente cambie las prioridades. Los desarrolladores también pueden proponer al cliente y ayudarlo a realizar la priorización. Esta técnica es popular y se la ha utilizado por años.

2.2.8 Modelado

El Modelado actúa como un puente entre el análisis y el diseño. Algunos métodos utilizan diferentes técnicas para formular y analizar requerimientos del sistema. Las técnicas más populares son Modelos de flujo de datos, modelos semánticos de datos y modelos orientados a objetos.

2.2.9 Documentación de Requerimientos

El propósito de documentar los requerimientos es el comunicar la comprensión de éstos entre clientes y desarrolladores. El documento de requerimientos explica el dominio de aplicación y el sistema a ser desarrollado. Éste puede ser la base para evaluar subsecuentes productos y procesos como casos de prueba, actividades de validación y verificación, así como también para el control de cambios.

Los documentos de ingeniería de requerimientos son largos. La mayoría están compuestos de cientos o miles de páginas; cada página contiene muchos detalles que pueden tener efectos profundos en el resto del sistema.

Normalmente, las personas se encuentran con dificultades para comprender documentos de este tamaño, sobre todo si lo leen cuidadosamente. Es casi imposible leer un documento de especificación de gran tamaño, pues difícilmente una persona puede memorizar los detalles del documento. Esto causa problemas y errores que no son detectados hasta después de haberse construido el sistema.

Un buen documento de requerimientos debe ser claro, sin ambigüedades, completo, correcto, entendible, consistente, conciso, y factible.

2.2.10 Validación de Requerimientos

La validación de requerimientos certifica que estos representan una descripción razonable del sistema a ser desarrollado. Las entradas para este proceso son: el documento de requerimientos, los estándares organizacionales, y el conocimiento organizacional. Las salidas para el proceso de validación son una lista de problemas reportados en el documento de requerimientos y las acciones tomadas entre desarrolladores y clientes para encararlos.

A través de la actividad de validación es recomendable realizar interrogantes como las que se presenta a continuación con la cualidad que estas buscan crear como producto del levantamiento de requerimientos y el desarrollo del producto mismo:

- ¿Están incluidas todas las funciones requeridas por el cliente? (completo)
- ¿Existen conflictos en los requerimientos? (consistencia)
- ¿Tiene alguno de los requerimientos más de una interpretación? (no ambiguo)
- ¿Está cada requerimiento claramente representado? (entendible)

- ¿Pueden los requerimientos ser implementados con la tecnología y el presupuesto disponible? (factible)
- ¿Está la SRS escrito en un lenguaje apropiado? (claro)
- ¿Existe facilidad para hacer cambios en los requerimientos? (modificable)
- ¿Está claramente definido el origen de cada requisito? (rastreable)
- ¿Pueden los requerimientos ser sometidos a medidas cuantitativas? (verificable)

La validación de requerimientos es importante pues de ella depende que no existan elevados costos de mantenimiento para el software desarrollado.

2.2.11 Evolución de requerimientos

Esta fase incluye todas las actividades que tienen que ver en las áreas de control de cambios, control de versión, monitoreo de requerimientos. El objetivo es capturar, almacenar, diseminar y manejar información.

Los requerimientos, a pesar que son una manera de comprender mejor las necesidades de los usuarios y los objetivos de la organización, pueden cambiar, por lo tanto, es esencial planear esos posibles cambios a los requerimientos cuando el sistema sea desarrollado y está siendo utilizado. La actividad de evolución es un proceso externo que ocurre a lo largo del ciclo de vida del software.

Los requerimientos cambian por diferentes razones. Las más frecuentes son:

- Porque al analizar el problema, no se hacen las preguntas correctas a las personas correctas.
- Porque cambió el problema que se estaba resolviendo.

- Porque los usuarios cambiaron su forma de pensar o sus percepciones.
- Porque cambió el ambiente de negocios.
- Porque cambió el mercado en el cual se desenvuelve el negocio.

El cambio a los requisitos involucra modificar el tiempo en el que se va a implementar una característica en particular, lo cual puede tener impacto en otros requerimientos. Por esto, la administración de cambios involucra actividades como establecer políticas, guardar históricos de cada requerimiento, identificar dependencias y mantener un control de versiones.

El versionamiento de requerimientos es tan importante como el que se realiza con el código por razones que tienen que ver con el cambio y evolución de un producto de software.

Entre algunos de los beneficios que proporciona el control de versiones están:

- Prevenir cambios no autorizados.
- Guardar revisiones de los documentos de requerimientos.
- Recuperar versiones previas de los documentos.
- Administrar una estrategia de versiones puestas en producción.
- Prevenir la modificación simultánea a los requisitos.

En vista que las peticiones de cambios provienen de muchas fuentes, las mismas deben ser enrutadas en un solo proceso. Esto se hace con la finalidad de evitar problemas y conseguir estabilidad en los requerimientos.

2.3 LA METODOLOGIA RUP

La metodología a utilizarse para este proyecto es RUP que significa Rational Unified Process. Este modelo está compuesto de las mejores prácticas de muchos desarrolladores a través de los años en una extensa variedad de situaciones.

La metodología provee una manera disciplinada de asignar y administrar tareas y responsabilidades dentro de una organización. Aplicando este proceso se puede desarrollar sistemas de alta calidad que satisfacen las necesidades de sus usuarios finales y todo eso dentro de un tiempo y presupuesto estipulados con anterioridad.

RUP guía efectivamente en la aplicación de técnicas modernas como el desarrollo iterativo, mitigar riesgos en cada etapa del proceso y una continua evaluación de la calidad del software. Aunque miles de proyectos están implementando RUP efectivamente, muchos de los equipos están intimidados porque creen que RUP es complejo y enorme. La verdad es que no debe ser enorme y no es complejo.

Muchos equipos todavía utilizan un proceso en cascada para el desarrollo de proyectos, donde se completa cada fase en una estricta secuencia de requerimientos, donde vienen el análisis y diseño seguidos a continuación de implementación/integración y al final pruebas. Esto lo que hace es dejar en espera a miembros claves del equipo de desarrollo por períodos extendidos de tiempo y mueven la etapa de pruebas para el final cuando los problemas tienden a ser más difíciles y costosos de resolver. Esto significa serios riesgos al proyecto y los plazos establecidos.

RUP en cambio utiliza un proceso iterativo que se resume en una secuencia de pasos incrementales o iteraciones. Cada iteración incluye algunas o la mayoría de las disciplinas de desarrollo (requerimientos, análisis, diseño, implementación, pruebas). Cada iteración tiene un

set bien definido de objetivos y produce una implementación parcial del sistema final. Además, cada iteración sucesiva se basa e incrementa los logros de las iteraciones previas para evolucionar y refinar el sistema hasta que el producto final está completo.

Las primeras iteraciones tendrán un mayor énfasis en requerimientos, análisis y diseño, mientras que las iteraciones posteriores tendrán un mayor énfasis en implementación y pruebas.



Figura 2.1 Iteraciones de la Metodología RUP

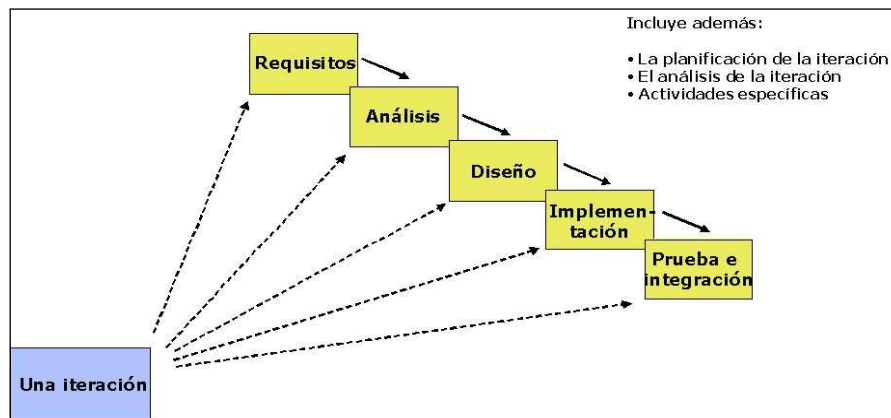


Figura 2.2: Fases de una Iteración

2.3.1 Concepto de RUP

RUP o Rational Unified Process es un framework para el desarrollo de software que consiste de uno a más sets completos de elementos de proceso para desarrollo de proyectos de software. RUP define un proyecto como un grupo de disciplinas, por ejemplo administración de requerimientos, implementación, etc. RUP apoya al proyecto de inicio a final a través del ciclo de vida de éste y por lo tanto se divide en varias fases. Un proyecto es ejecutado por un número de actores, cada uno con roles definidos. Cada rol participa en una o más actividades para producir artefactos. Una disciplina puede correr iteraciones o repeticiones en una fase. Actividades, roles y artefactos son los elementos fundamentales de esta metodología.

2.3.2 Artefactos

Los artefactos constituyen entregas que se realiza en lo que se refiere a documentación, prototipos, etc. La producción de artefactos se puede dar a cualquier instancia del proyecto y como resultado de iteraciones.

Entre los artefactos principales de todo proyecto RUP tenemos los siguientes:

2.3.3 La visión

La visión ayuda a los integrantes del equipo de desarrollo a comprender que construir y más tarde les ayuda a darse cuenta que tareas han terminado. La visión captura la esencia del flujo de requerimientos. Es una visión de alto nivel del sistema que permite dar al lector un entendimiento claro del sistema que se está desarrollando.

2.3.4 La lista de riesgos

Cuáles son los riesgos que se presentan y como se los mitiga. Cuando se habla de riesgos, se debe considerar los siguientes aspectos: personas, procesos, y herramientas. Es preciso considerar riesgos que son reales, que tienen una probabilidad aceptable de ocurrir y por este medio también se los debe priorizar de acuerdo a la probabilidad de ocurrir y a los costos asociados con ellos. La lista de riesgos no es definitiva ya que durante el proyecto mientras aparecen más riesgos, se la puede cambiar.

2.3.5 Los casos de uso

Los casos de uso definen las interacciones de los usuarios con el sistema o sistemas. El usuario también se llama actor que produce resultados tangibles. La realización de casos de uso asegura que después de cada iteración se produzca algo de valor. Los casos de uso son muy importantes ya que son la base de los requerimientos. También se puede realizar casos de uso para requerimientos no funcionales, que son aquellos que no son basados en casos de uso principales sino para otros propósitos como tiempo de respuesta, etc.

La importancia de los casos de uso no sólo se refleja en el levantamiento de requerimientos sino también en los planes de pruebas, documentación, desarrollo, entre otros.

Para la representación de los casos de uso se puede decidir la utilización de UML, texto o ambos. Las dos alternativas son buenas pero eso depende de cada equipo y el grado de formalidad que se va a aplicar en el proyecto. No siempre se necesita casos de uso, por ejemplo si se está desarrollando una aplicación como un driver o un compilador.

2.3.6 Pruebas

Cuando se utiliza RUP se puede comenzar realizando pruebas desde que se completa los primeros casos de uso. De esta manera se puede asegurar que lo que se va a producir cumple con los requerimientos de manera efectiva.

2.3.7 La arquitectura

La arquitectura es una parte esencial de un proyecto pero puede expresarse de una manera informal; algunos equipos entregan la primera versión sin una arquitectura formal, y desde la segunda versión del software recién empiezan a planear y documentar sobre la arquitectura a utilizarse. Otros en cambio desarrollan una arquitectura desde el comienzo. Lo importante es darse cuenta que sea al comienzo o al final, se tiene una arquitectura bien diseñada.

2.3.8 El plan de desarrollo

El plan de proyecto diseña las iteraciones y cronogramas así como las iteraciones, riesgos durante la fase de elaboración. Éste ayuda a reducir la probabilidad de sorpresas técnicas o tareas repetitivas y no optimas en las etapas avanzadas del proyecto. De acuerdo a la experiencia obtenida en el campo y en transcurso de muchos proyectos por varios desarrolladores en distintas situaciones y lugares, se puede sugerir una fase de Inicio con una iteración temprana que determine objetivos muy simples para ayudar a los miembros del equipo a trabajar juntos, de esta manera, las subsecuentes reuniones se encargaran de aspectos relacionados con riesgos al proyecto

2.3.9 Proyectos Pequeños y RUP

Al pensar en un proyecto pequeño nos viene a la mente el número de personas y es lo correcto ya que se considera generalmente un proyecto pequeño al compuesto por diez o menos personas. Pero también existen otros parámetros de medición como:

- La duración del proyecto
- La cantidad de código producido
- La cantidad de dinero invertido
- El nivel de complejidad del código
- La cantidad de código producido por la aplicación

De aquí los más principales que se pueden destacar y ampliar de cierto modo y que definen claramente un proyecto pequeño son:

- El número de personas envueltas en el proyecto lo que es relacionado con el **nivel de formalidad del proyecto.**
- La complejidad del nuevo código, que también se correlaciona con el número de personas en el proyecto.
- La duración del proyecto

Hemos puesto énfasis en el nivel de formalidad que tiene que ver con el número de artefactos que se entrega como resultado del proyecto ya que por ejemplo, si se trabajara en un proyecto de alta complejidad con personas en una distribución geográfica amplia que llegarán a las 50, 100 o 200, el nivel de formalidad debería por naturaleza ser alto. Si en cambio tenemos un

equipo de diez personas o menos trabajando en la misma oficina en un sistema de poca complejidad, el nivel de formalidad cae a un mínimo. Se puede también tener casos de sistemas con pocas personas pero de alta complejidad y por lo tanto formalidad, pero esto no es la regla. RUP tiene muchas iteraciones que también se las considera como mini proyectos.

Con muchos mini proyectos que son parte de un grande, tendremos mucha documentación y formalidad, pero que es más importante, la formalidad o el producto que realmente funcione y satisfaga las expectativas y necesidades del cliente? La respuesta debería ser la segunda, ya que la formalidad tiene como primordial propósito el ayudar a un mejor desarrollo y mantenimiento del sistema más el usuario, que no es técnico y no sabe de desarrollo de software en la gran mayoría de los casos, no le interesa y prefiere saltarse esos niveles de formalidad para apreciar un producto que trabaja y le sirve con documentación simple que él y sus usuarios críticos puedan entender en el menor tiempo posible. Además el cliente debe tener la capacidad de evaluar el producto y ver su progreso.

En mención de lo anterior, se considerará para el presente proyecto mantener la cantidad de artefactos acorde a las demandas del proyecto, dejando de lado la excesiva formalidad que puede originarse si no se pone una tecnología como RUP en práctica correctamente.

2.3.10 Beneficios para proyectos pequeños

Los beneficios de la metodología para proyectos pequeños se pueden resumir de la siguiente manera:

- RUP provee información y guía de las mejores prácticas de desarrollo de software probadas en miles de proyectos en todo el mundo.

- RUP ayuda a comprender las actividades, roles, artefactos que se necesita en un proyecto y especifica cuando es mejor usar cada uno.
- RUP brinda información completa y detallada para efectivamente aplicar técnicas diversas en un proyecto. Puede también ayudar con el uso de herramientas Rational y el lenguaje UML.
- RUP también puede ayudar a ajustar el proceso para enfocarse en asuntos o áreas específicas dentro del proyecto, por ejemplo en el área de administración de requerimientos.

2.4 Fases del RUP

2.4.1 Inicio o Conceptualización

La fase de inicio es muy significativa para los nuevos proyectos en donde se debe apuntar y detectar riesgos al negocio y al proyecto antes de proceder con cualquier actividad. Para proyectos que se enfocan en mejoras a un sistema existente, la fase de Inicio es más corta pero se enfoca igualmente en asegurar que el proyecto vale la pena ser realizado así como la factibilidad de su desarrollo.

Durante esta fase, se realiza un caso de negocio, el cual ayudará a construir el software. La Visión es un artefacto clave que es producido en esta fase del RUP y consiste en una descripción a alto nivel del sistema.

La Visión puede ser breve, quizá un párrafo o dos. A veces la Visión contiene las características críticas que el sistema debe brindar al cliente. A continuación un ejemplo tomado de la Visión de Rational Software

“Para impulsar la position de Rational como un líder mundial en desarrollo de herramientas, servicios y mejores prácticas, por medio de la mejora de las relaciones con el cliente a través de una presencia dinámica y personalizada en el Web brindando al visitante servicio, apoyo y contenido enfocado. El nuevo proceso y tecnología impulsarán a los proveedores de contenido a acelerar las labores de publicación y calidad del contenido a través de una solución automatizada simplificada.”

La Visión puede ser representada en un documento formal o de manera informal. Su propósito es poner a todos en ruta hacia un objetivo. La Visión es algo sumamente poderoso, hablando de proyectos. A continuación la descripción de la Visión según el libro “Software para su cabeza” por Jim y Michele McCarthy.

“El modelo para la Visión es más que solo una oración, ya que incluye características de alto nivel del producto. Así que cuando hablamos de la Visión como artefacto, queremos decir una oración que describa la Visión y otra que describa las características”

Existen cuatro actividades principales en esta fase de Inicio según RUP son:

Formular el alcance del proyecto

Si vamos a producir un sistema debemos saber qué se va a hacer y cómo satisfará al cliente. En esta actividad capturamos el contexto y los requerimientos más importantes con suficiente detalle para derivar criterios de aceptación para el producto.

Preparar el ambiente para el proyecto

Cualquier proyecto necesita un ambiente para su desarrollo y pruebas. Sea que se utilice técnicas de programación en pares o técnicas más tradicionales para determinar los recursos físicos, herramientas de desarrollo y procedimientos que el equipo debe seguir, se lo hace en la fase de Inicio.

Como se puede ver, no toma mucho tiempo el aplicar la fase de Inicio en un proyecto pequeño, en algunos casos es posible hasta realizarla en un par de días.

Durante la fase de Inicio, no se resuelve todos los problemas que el desarrollador anticipa, más bien solo se quiere obtener una idea del trabajo a realizarse creando algo como un mapa de caminos. Cuando se empieza a planificar el viaje, no se necesita saber la ruta exacta por la cual se viajará, pero se sabe de seguro la dirección y los lugares que se visitará así como el tiempo aproximado que el viaje tomará. La fase de Inicio es eso exactamente.

2.4.2 Elaboración

Cuadro 3.1: Fase de Elaboración

Es importante educar a los clientes sobre los objetivos de las fases, especialmente la de Elaboración, en la cual el objetivo es atacar el riesgo y no necesariamente dar funcionalidad. El progreso durante la fase de elaboración debe ser medido por el número de riesgos considerados y atacados antes que por el número de casos de uso implementados. Aunque lo anterior es una métrica importante, esta es utilizada en la fase de Construcción donde el objetivo es implementar funcionalidad.

La palabra Elaboración significa refinamiento o desarrollo minucioso. En la fase de Elaboración es donde más se nota la diferencia en la metodología de cascada y el proceso iterativo y una importante diferencia radica en las actividades de la fase.

El propósito de la fase de elaboración es analizar el dominio del problema, establecer una arquitectura apropiada para el sistema, desarrollar el plan de proyecto y eliminar los mayores elementos de riesgo del proyecto.

Para lograr estos objetivos se debe tener una visión amplia aunque poco profunda del sistema.

La pregunta clave que nos debemos hacer durante esta fase es **PODEMOS CONSTRUIRLO?** Si se tiene dificultad respondiendo esta pregunta significa que se puede tener problemas más adelante porque se está empujando al riesgo hacia el final del proyecto.

Las decisiones con respecto a la arquitectura deben ser realizadas con un entendimiento aceptable de todo el sistema: Su alcance, sus requerimientos funcionales y no funcionales.

Es fácil decir que la fase de elaboración es la más crítica de las cuatro fases ya que al final de esta se empezará la construcción y la transición. Aunque el proceso debe siempre acomodarse a los cambios, la fase de elaboración asegura que la arquitectura, los requerimientos y los planes son lo suficientemente estables además que los riesgos mitigables para así tener la capacidad de determinar el costo y el cronograma de realización del desarrollo , aspectos claves para la siguiente fase .

En la fase de elaboración, un prototipo ejecutable de la arquitectura es construido en una o más iteraciones dependiendo siempre en el alcance del proyecto, su tamaño y complejidad.

Este esfuerzo debe al menos considerar los casos de uso más críticos que fueron identificados en la fase de inicio, para así exponer los mayores riesgos tecnológicos del proyecto.

El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

Los objetivos de esta fase son:

Definir, validar y cimentar la arquitectura.

Completar la visión.

Crear un plan fiable para la fase de construcción. Este plan puede evolucionar en sucesivas iteraciones. Debe incluir los costes si procede.

Demostrar que la arquitectura propuesta soportará la visión con un coste razonable y en un tiempo razonable.

Al terminar deben obtenerse los siguientes resultados:

- Un modelo de Casos de Uso completa al menos hasta el 80%: todos los casos y actores identificados, la mayoría de los casos desarrollados.
- Requisitos adicionales que capturan los requisitos no funcionales y cualquier requisito no asociado con un Caso de Uso específico.
- Descripción de la arquitectura software.
- Un prototipo ejecutable de la arquitectura.
- Lista de riesgos y caso de negocio revisados.
- Plan de desarrollo para el proyecto.
- Un Documento SAD (Software Architecture Description)
- Un caso de desarrollo actualizado que especifica el proceso a seguir.
- Un manual de usuario preliminar (opcional).

En esta fase se debe tratar de abarcar todo el proyecto pero sin ahondar demasiado. Sólo se profundiza en los puntos críticos de la arquitectura o riesgos importantes.

En la fase de elaboración se actualizan todos los productos de la fase de inicio.

Los criterios de evaluación de esta fase son los siguientes:

- La visión del producto es estable.
- La arquitectura es estable.
- Se ha demostrado mediante la ejecución del prototipo que los principales elementos de riesgo han sido abordados y resueltos.
- El plan para la fase de construcción es detallado y preciso. Las estimaciones son creíbles.
- Todos los interesados coinciden en que la visión actual será alcanzada si se siguen los planes actuales en el contexto de la arquitectura actual.
- Los gastos hasta ahora son aceptables, comparados con los previstos.

Si no se superan los criterios de evaluación quizá sea necesario abandonar el proyecto o replanteárselo considerablemente.

Fase de Elaboración Iteraciones

En la fase de Elaboración muchos riesgos son mitigados al producir una arquitectura con una muestra ejecutable, eso es , un conjunto de los aspectos más esenciales del sistema que permiten , muy concretamente, demostrar capacidades clave y así asegurarse que los riesgos son mitigados. Si se ha construido un sistema con la misma tecnología que se va a usar en un proyecto anterior, se puede alcanzar los objetivos de esta fase por medio de una sola iteración

porque existe una cantidad limitada de riesgos que atacar. Se puede reutilizar soluciones del pasado y así hacer un progreso rápido.

Pero si no se es lo suficientemente experimentado en el dominio de la aplicación, si el sistema es muy complejo o si se está utilizando nueva tecnología, se puede necesitar dos o más iteraciones para obtener la arquitectura correcta y mitigar eficazmente los riesgos. Otros factores que pueden llevar al hecho de tener varias iteraciones incluyen el realizar un desarrollo distribuido, tener muchos clientes o complejos arreglos contractuales, la necesidad de cumplir con regulaciones de seguridad u otros estándares externos.

En la fase de elaboración se pone un énfasis especial en los siguientes objetivos:

Obtener un entendimiento más claro de los requerimientos

Diseñar , implementar, validar , y cimentar la arquitectura

Mitigar riesgos esenciales y producir un cronograma más exacto así como los estimados de costos.

Refinar el caso de uso de desarrollo y establecer el ambiente de desarrollo.

Objetivo 1. Obtener un entendimiento más claro de los requerimientos

Hacia el final de la fase de Inicio se debió haber producido un buen documento de Visión , así como una descripción detallada del 20 por ciento de los casos de uso más esenciales y una breve descripción de los restantes casos de uso.

Al final de la fase de Elaboración, se debe tener una descripción completa de la mayoría de los casos de uso. Algunos pueden ser simples o similares a otros casos de uso pero que operan con diferentes datos, que se los puede tranquilamente posponer hasta la fase de Construcción, o

aun no mencionarlos formalmente. Detallar estos casos no aporta la detección de ningún riesgo.

Objetivo 2: Diseñar, Implementar, Validar y Delimitar la Arquitectura

La arquitectura de Software es algo muy importante que debe ser realizado cuidadosamente ya que existen varias opciones y es mejor escoger la más adecuada. Simplifiquemos este concepto a unas pocas decisiones de diseño que deben ser tomadas en cuenta:

Los bloques más importantes del sistema y sus interfaces , así como también la decisión de construir, comprar, reutilizar algunos de esos bloques constitutivos.

Una descripción de cómo estos bloques interactuarán .en producción para implementar los más importantes escenarios que se ha podido identificar.

Una implementación y pruebas de un prototipo de esta arquitectura para validar que está en efecto trabaja y que los mayores riesgos técnicos se encuentran resueltos además de tener atributos de calidad apropiados: en ejecución, escalabilidad, y costo.

En RUP , una arquitectura no es limitada a un dibujo en un papel o planos . Para validar una arquitectura, se necesita más que una representación de esta en papel , se necesita una arquitectura ejecutable que puede ser probada para verificar que cumple con los objetivos y se encarga de las necesidades de los clientes. Esto constituye la base para la realización del resto del sistema.

Cualquier arquitectura ejecutable es una implementación parcial del sistema construido para demostrar que el diseño de la arquitectura será capaz de soportar la funcionalidad clave y más

importante aún exhibir las propiedades correctas en términos de ejecución , capacidad, confiabilidad , escalabilidad , entre otros.

Al final de la fase de Elaboración, se define la arquitectura, lo que significa que esta será una referencia estable para construir el resto del sistema.

La arquitectura ejecutable es construida como un **prototipo evolucionario**, con la intención de retener capacidades validadas y aquellas con una alta probabilidad de satisfacer requerimientos del sistema cuando la arquitectura haya madurado.

Al final de la fase de Elaboración, se tendrá una arquitectura que será una referencia estable para construir el resto del sistema.

Desde este punto en adelante, se debe modificar la arquitectura con cuidado, y si se debe hacerlo , debe ser por una buena razón ya que de esto depende la estabilidad del equipo de desarrollo. Mientras más grande es el equipo de desarrollo y mas técnicamente complejo es el proyecto, más importante es tener una arquitectura concreta para el final de esta fase. En contraste, mientras más pequeño es el equipo de desarrollo y menos complejo es el proyecto , existe más libertad de modificar la arquitectura.

Al final de la fase de Inicio se tenía una idea de los riesgos que se podía encontrar , especialmente en las áreas de adquisición de tecnología y recursos reutilizables como frameworks existentes , software , etc. Pero en esa instancia muchas preguntas quedaron sin contestarse para ser finalizadas durante la Elaboración.

Por todas estas razones, al comienzo de la fase de Elaboración, se debe tener un entendimiento casi totalmente claro de la clase de sistema que se está yendo a construir. Antes que inventar una arquitectura, se debe divisar el uso de un framework quizá uno comercial o uno que haya tenido éxito con previos proyectos. Si no se desea proceder de esa manera, se necesita identificar los bloques que pueden ser subsistemas y componentes. Por ejemplo: Un sistema de transacciones online típicamente requiere un componente o subsistema que maneja cada bloque importante: Carrito de compras, Pagos online , promociones, adquisiciones, etc. Para cada uno de estos bloques se debe describir las capacidades o características clave que estos deben ofrecer como sus interfaces hacia el resto del sistema.

En paralelo, cuando se identifica esos subsistemas y componentes, se necesita evaluar recursos dentro y fuera de la compañía. Se puede adquirir un conjunto de componentes que implementan el concepto del Carrito de compras?. Estos componentes satisfacen nuestras necesidades??. Cuáles son las implicaciones legales y financieras??. Tienen estos componentes la capacidad de expansión que vaya acorde a nuevas tecnologías y requerimientos de los usuarios??. Se tiene acceso al código fuente para realizar los cambios necesarios???

Está el código apropiadamente documentado con guías acerca del diseño de los componentes y como se los utiliza y se los prueba??.

Como recomendación, se debe utilizar casos de uso que tengan significancia dentro del contexto de la arquitectura, ya que estos son los que la conducirán a través de las fases posteriores. También se necesita identificar ciertos elementos dentro de los requerimientos, posiblemente requerimientos no-funcionales, que son difíciles, desconocidos o riesgosos, y

encontrar casos de uso que ilustren los puntos difíciles cuya implementación forzaría una confrontación y resolución del riesgo.

Para algunos sistemas, uno o dos casos de uso pueden constituir el centro o core de la aplicación, los demás son casos de uso de apoyo.

Objetivo 3 : Mitigar riesgos esenciales y producir cronogramas y costos exactos.

Durante esta fase se mitiga una amplia variedad de riesgos técnicos siendo estos riesgos relacionados con el entendimiento de los requerimientos de los usuarios , así como también los riesgos de poner el proyecto en marcha en el ambiente de producción.

Planificación del proyecto y estimación de los costos

La fase de Elaboración permite tener una información más exacta que nos permite actualizar nuestro plan de proyecto y el costo estimado.

- Se tiene requerimientos detallados y eso ayuda a comprender mejor que tipo de sistema se está construyendo y siendo de este modo es fácil realizar las debidas actualizaciones a la Visión del negocio.
- Se ha implementado una arquitectura ejecutable del sistema , lo que significa que se ha resuelto los problemas más difíciles y esto nos presenta una área bien definida en la que vamos a trabajar.
- Se ha mitigado la mayoría de riesgos. Esto radicalmente ayuda a realizar un trabajo más eficiente al realizar los cronogramas y estimado de costos.
- Se entiende ahora cuan efectivamente se está trabajando con el personal, las herramientas y la tecnología porque se ha utilizado estos tres elementos para ir a través

del ciclo de vida completo al menos una vez (una vez por cada iteración de la fase de Elaboración).

Objetivo 4: Refinar el caso de desarrollo y establecer el ambiente de desarrollo

Durante el Inicio se definió que procesos a seguir y se documentó el **caso de desarrollo**.

También se definió que herramientas utilizar.

En la fase de Elaboración, se realizó el ciclo de vida completo que incluye diseño, implementación y pruebas de la arquitectura.

Para apoyar estas actividades, se completa la instalación y despliegue del proceso , de este modo se aprende que trabaja bien para el proyecto y que no trabaja. Se actualiza el caso de desarrollo de acuerdo a estas actividades.

Hito de la Fase: Arquitectura del proyecto

Al final de la fase de Elaboración está el Hito de la Fase que en este caso es la Arquitectura misma del proyecto. En este punto analizamos los objetivos detallados del sistema así como el alcance del proyecto, la elección de arquitectura y la resolución de los mayores riesgos. Si el proyecto no llegara a este hito, sería cancelado o al menos seriamente reconsiderado. Es mejor que esto pase temprano en el proyecto antes que más tarde. El enfoque iterativo mas la combinación de este hito nos forza a realizar tal decisión.

La revisión del Hito incluye los siguientes criterios de evaluación:

- Está la Visión y los requerimientos estables?
- Es la arquitectura estable??
- Son las estrategias de pruebas y evaluación probadas?

- Habiendo probado y evaluado los prototipos ejecutables , se demostró que los elementos de mayor riesgo han sido tratados y resueltos
- Son los planes de iteración para la fase de Construcción lo suficientemente detallados para permitir que el trabajo continúe??
- Están los planes de iteración para la fase de Construcción apoyados por estimados creíbles??
- Es nuestro gasto actual de recursos aceptable , comparado con el nuevo??

2.4.3 Construcción

La construcción se refiere al desarrollo eficaz y eficiente en términos de costo y tiempo de un producto completo, que es una versión operacional que puede ser desplegada en la comunidad de usuarios. Esto se traduce a los siguientes objetivos:

Objetivo 1: Minimizar los costos de desarrollo y llegar a tener algún grado de paralelismo en el trabajo de los equipos de desarrollo.

Optimizar recursos y evitar actividades innecesarias. Aun los proyectos pequeños generalmente tienen componentes que pueden ser desarrollados independientemente, permitiendo que exista ese grado de paralelismo entre desarrolladores y equipos de desarrolladores.

Iterativamente desarrollar un producto completo que esté listo para ser presentado a la comunidad de usuarios

Desarrollar la primera versión operacional del sistema (release beta) describiendo los casos de uso restantes y otros requerimientos, llenando los detalles del diseño aun no completos, concreción de la implementación y pruebas del software. Así también se debe determinar si el software, los sitios, y los usuarios están todos listos para el despliegue de la aplicación.

Fase de Construcción Iteraciones

El numero de iteraciones requeridas para la fase de Construcción varia de proyecto a proyecto, pero en la mayoría de los casos, esta fase tiene más iteraciones (usualmente cuatro) que las otras fases del RUP.

Así la pregunta es : Que va dentro de estas iteraciones??. La planificación de las iteraciones es ampliamente manejado por las partes de los casos de uso que deben ser implementados. Se trata aquí de implementar los casos de uso más críticos y relevantes para los clientes, así como aquellos asociados con el mayor riesgo técnico. Especialmente en la primera iteración, y algo de la segunda, empezar con una implementación parcial de los casos de uso (eso es , implementar únicamente algunos de los escenarios dentro del caso de uso) , con el objetivo de alejar la mayor cantidad de riesgo que se pueda y obtener una implementación razonable antes de detallar los casos de uso.

Una vez que se haya decidido cuales casos implementar, o parcialmente implementar, se identifica que componentes necesitan colaborar para proveer la funcionalidad del caso de uso. Estos son los componentes que deben ser diseñados, implementados y probados dentro de la

iteración. Esta identificación provee con un mejor entendimiento del tiempo requerido para implementar los casos de uso basado en los recursos disponibles, el alcance del trabajo necesita ser cambiado para la iteración.

Comunicación dentro de la fase de Construcción

La organización alrededor de la arquitectura minimiza la sobrecarga de comunicación...

Los caminos de comunicación entre los miembros del equipo crecen **geométricamente** con el tamaño del equipo. La organización alrededor de la arquitectura radicalmente reduce el número de caminos de comunicación dentro de un equipo. Los asuntos concernientes a la interacción con otros subsistemas son resueltos por el equipo de arquitectura, el cual es custodio de las interfaces entre subsistemas

Un incremento en los caminos de comunicación destruye la eficiencia de un equipo, así que la necesidad es imperativa para hallar e implementar medios de comunicación más óptimos. Esto se puede alcanzar teniendo un responsable para la arquitectura y varios equipos pequeños, cada uno responsable de cada subsistema.

La comunicación entre estos pocos equipos es manejada a través de la organización que se haya puesto a funcionar alrededor de la arquitectura.

Planificación de la Integración

Con el desarrollo iterativo, se hace más complejo planificar builds y pruebas de integración. Cada iteración necesita un plan de Integración especificando cuales características deben ser

probadas en cada build y cuales componentes necesitan ser integrados para producir las características requeridas, tales como, los casos de uso, partes de los casos de uso, u otras funcionalidades. Las pruebas pueden incluir: pruebas de funcionalidad, carga, stress , etc.

En muchos casos, un build es producido incrementalmente integrando compilaciones más pequeñas. Esto es típicamente realizado de abajo hacia arriba. Para cada build, se define cuales subsistemas entran en esta operación y que otros subsistemas deben estar disponibles como stubs (código que emula las características requeridas del subsistema).

Hacer cumplir la arquitectura

Para obtener el máximo beneficio del trabajo realizado con la arquitectura, se necesita hacerla cumplir tanto para equipos pequeños como grandes.

Se ha definido mecanismos de arquitectura, eso es, soluciones reutilizables para problemas comunes tales como, persistencia, comunicación entre procesos , etc. Ahora se necesita prevenir que cada desarrollador arbitrariamente invente nuevas soluciones para estos problemas.

Por eso es que se va a entrenar a los desarrolladores en la arquitectura y los mecanismos de arquitectura disponibles, acoplados a la función de los desarrolladores, es así que los arquitectos y desarrolladores toman parte en este punto.

También hay que asegurarse que las interfaces a los subsistemas no sean arbitrariamente cambiadas y que cualquier cambio sea comunicado a otros equipos para minimizar el impacto en ellos. Esta comunicación puede ser realizada por medio de un CM (Configuration Management System: Sistema de Manejo de Configuraciones).

Asegurar progreso continuo

Para asegurar progreso continuo, se necesita establecer metas a corto plazo y continuamente probar que se las ha cumplido.

Las siguientes guías son recetas de éxito probadas:

- **Crear un equipo con una misión:** Se debe por todos los medios evitar equipos orientados a funcionalidades, donde los analistas están organizados en un equipo y lanzan los requerimientos sobre la pared a los desarrolladores quienes los diseñan e implementan, y estos a su vez, lanzan el código sobre la pared hacia los desafortunados probadores quienes se topan con la difícil tarea de descubrir que tiene que ser probado. Una mejor practica y que no es receta para el desastre es tener equipos que trasciendan todas las funcionalidades donde cada miembro es responsable por la aplicación y por ende ,el progreso del equipo. Es recomendable también tener una rápida reunión diaria para discutir el status de los proyectos y decidir en que enfocarse ese día.
- **Poner objetivos claros y alcanzables para los desarrolladores**

Cada desarrollador debe tener un panorama claro de lo que debe alcanzar en una iteración dada o dentro de dicha iteración. Los desarrolladores tienen también su voto con respecto a la posibilidad de realizar las tareas dentro del tiempo prescrito y el grado de dificultad que estas presenten. No es cuestión de imponer al desarrollador sino de consensuar enfatizando la importancia que tiene la labor de cada uno de ellos para el progreso del proyecto.

- **Continuamente demostrar y probar el código**

Mida el progreso primariamente evaluando que código es compilado y probado. No se quede satisfecho por las típicas aseguraciones que dicen “estamos en un 90 por ciento” . La demostración y pruebas continuas del código ejecutable es la única manera de asegurar progreso.

- **Forzar la integración continua**

Si es posible, es recomendable realizar builds diarios. Para proyectos grandes, o proyectos sin sistemas adecuados de manejo de configuraciones (CM) , los builds diarios no son posibles , así que se puede realizar con un tiempo más espaciado pero eso si frecuentemente. Esto nos asegura que tengamos siempre una retroalimentación sobre el funcionamiento del nuevo código en la aplicación.

Objetivo 2: Iterativamente desarrollar un producto completo que esté listo para su transición a la comunidad de usuarios

Describir los casos de uso restantes y otros requerimientos

Mientras se implementa y prueba un caso de uso , a menudo se requiere visitar algunos de los requerimientos detallados, y en muchos casos, se puede aun reconsiderar el caso de uso entero , al llegar a divisar mejores soluciones.

Diseño de la Base de Datos

Durante la fase de Elaboración se hace un borrador de lo que es la base de datos. En la fase de Construcción, este borrador es depurado, quizá se necesita más columnas en las tablas, vistas necesitan ser creadas, índices deben ser optimizados , etc. Como se puede ver , estos cambios

no son reestructuraciones mayores , que se debe evitar. Si se da esta necesidad, es un claro indicador de que la arquitectura no está lo suficientemente estable y que el comienzo de la fase de Construcción fue prematuro.

Implementación y Pruebas de Unidad

La planificación de una iteración es determinada por cuales casos de uso se debe implementar y probar así como cuando se debe hacerlo. La implementación de casos de uso es hecha componente por componente. Generalmente, al momento de llegar a la fase de Construcción, algunos de los componentes han sido ya implementados o parcialmente implementados.

Los desarrolladores necesitan probar sus implementaciones continuamente para verificar que el sistema se comporte como ellos esperan. Para probar los componentes, puede que sea necesario diseñar stubs y otros programas para la interacción con los componentes. Un ide visual puede utilizarse para generar estos programas. Una vez que se tienen listos los stubs, se puede correr un número de escenarios de prueba.

Usualmente, los escenarios de prueba se derivan de escenarios de casos de uso.

Para incrementar la calidad, se debe integrar continuamente y probar el sistema. Para minimizar los costos de pruebas, se necesita automatizar pruebas de regresión las cuales deben ejecutarse diariamente o semanalmente con el objetivo de poder fácilmente y rápidamente detectar defectos. Los siguientes pasos son una ayuda para el proceso de pruebas.

- Identificar lo que se va a probar, analizando el plan de iteración para asegurarse que se pruebe lo que ha sido generado en la actual iteración.

- Identificar ideas para las pruebas: es recomendable tener una lista de ideas que identifiquen que tipos de pruebas son apropiados para la iteración. Estas ideas son extraídas de fuentes como : La lista de riesgos, peticiones de cambio a la aplicación , casos de uso , u otros artefactos , modelos UML , etc.
- Analizar las ideas para las pruebas y seleccionar un subconjunto para probar. Se define entradas , salidas , condiciones de ejecución , y puntos de observación y control. Al analizar los casos de pruebas, se identifica la arquitectura para automatización de las pruebas, incluyendo la estructura general de componentes esenciales y scripts.
- Implementar las pruebas (manuales o automatizadas) para cada caso de prueba. Se organiza los test en suites y luego se los ejecuta.
- Analizar las fallas en las pruebas , si existen, se debe ingresar defectos y peticiones de cambio.

Despliegues tempranos y Ciclos de retroalimentación

Al ejecutar builds frecuentes, se fuerza una continua integración y verificación del código que funciona. Es crucial obtener una retroalimentación temprana sobre si la aplicación es útil, si provee el comportamiento necesario, exponiendo la aplicación a usuarios reales quienes pueden dar su opinión.

Los usuarios futuros del sistema puede que no deseen o no puedan dar tiempo a versiones tempranas del software. Por ejemplo, puede ser difícil convencer a cualquier usuario a que pase el día brindándonos retroalimentación , ya que los beneficios a esta etapa no son obvios

para el usuario. Esto es a menudo el caso cuando se construye productos comerciales, cuando la identidad de los futuros usuarios es desconocida.

Durante las etapas tempranas de la fase de Construcción puede que la aplicación sea difícil de utilizar, un dolor de cabeza el instalar. Esto es razón para tener un rol activo en las pruebas.

Basado en las necesidades de retroalimentación y disponibilidad de clientes , se puede considerar lo siguiente:

- Traer unos pocos usuarios al área de desarrollo y demostrar algunas características claves del software.
- Traer unos pocos usuarios al área de desarrollo y hacerles que utilicen el producto por algún tiempo.
- Instalar el software en un lugar de pruebas sentándose con los usuarios mientras estos realizan las pruebas.

Los beneficios de un despliegue de aplicación temprano no se hacen esperar ya que vienen en forma de retroalimentación concerniente al cumplimiento de los requerimientos o si necesitan ser cambiados. Esta retroalimentación cubre usabilidad, rendimiento, y la identificación de características insuficientes.

El probar en un ambiente de desarrollo que es diferente al de producción puede producir resultados diferentes y confusos. Las organizaciones que se enfocan en un control de calidad estricto, deben invertir en ambientes separados para este propósito.

Prepararse para el despliegue de la versión Beta

Un despliegue beta es una prueba “prerelease”, en el cual una muestra de los usuarios finales prueba el software.

El despliegue de la versión Beta se realiza al final de la fase de Construcción y es el principal enfoque de la fase de Transición.

Las pruebas de la versión Beta sirven dos propósitos:

- Probar la aplicación a través de la implementación actual.
- Proveer una vista preliminar del software.

Es importante tener usuarios novatos y avanzados, algunos de estos en diferentes ambientes y con diferentes necesidades. Esta variedad permite asegurar que los productos sean adecuadamente probados. Es también esencial que el producto este completo, basado en el manejo del alcance que ha ocurrido durante las iteraciones. Aunque todas las características deben ser implementadas, es aceptable tener algunos asuntos de calidad no resueltos tales como algún elemento inestable, archivos de ayuda o ventanas de mensajes que no son la versión final o una implementación parcial de una opción raramente utilizada. Para esto se necesita incluir instrucciones de instalación, manuales de usuario, tutoriales, y material de entrenamiento, o no se tendrá la debida retroalimentación de los probadores beta en estos materiales.

2.5.3 Transición

La versión del producto en la fase de Transición, no es necesariamente la versión final del este ya que todavía requiere de ajustes en funcionalidad, rendimiento y calidad.

El objetivo de la fase de Transición es asegurar que el software completamente cubra las necesidades de sus usuarios. La fase de Transición normalmente se compone de una o dos iteraciones que incluyen probar el producto en preparación para su release y realizar ajustes menores basados en la retroalimentación brindada por los usuarios. En este punto del ciclo de vida, todos los asuntos estructurales mayores deben haber sido pulidos, y la retroalimentación del usuario debe enfocarse únicamente en ajustes de configuración, instalación, asuntos de operación. Los proyectos más complejos pueden tener más iteraciones en la fase de Transición , con cada iteración produciendo una versión más refinada del producto. A menudo , en esta etapa ocurre que se deba completar alguna característica que ha sido pospuesta para cumplir un plazo anterior.

Cabe recalcar que la Transición difiere totalmente del desarrollo tradicional , primeramente porque se ingresa a la fase con una versión del sistema razonablemente estable, integrada y probada. A diferencia, en la metodología tradicional en cascada, la fase final de integración comienza con una ruptura mayor , a veces ni siquiera se puede compilar el sistema, las interfaces entre subsistemas no son compatibles o el sistema se cuelga frecuentemente , resultando en un mega esfuerzo extra para corregir esos problemas y una demora de varias semanas antes que el sistema pueda correr correctamente.

Objetivos de la fase de Transición

La fase de Transición tiene los siguientes objetivos:

- **Conducir pruebas beta para validar que las expectativas de los usuarios han sido cumplidas**

Esto típicamente requiere algunas actividades de ajuste tales como: arreglo de bugs y realizar mejoras al rendimiento y la facilidad de operación.

- **Entrenar usuarios y personal de mantenimiento para que logren ser autosuficientes**

Estas actividades aseguran que la organización que adopta el sistema están calificados para utilizar el sistema y han movido los datos de sistemas antiguos al sistema nuevo o tomado otras medidas para que el nuevo sistema opere exitosamente.

- **Preparar el sitio del despliegue y preparar bases de datos**

Para que el sistema corra sin problemas, puede que haya que adquirir nuevo hardware, adecuar el ya existente o convertir datos de un sistema anterior al nuevo.

- **Preparación de empaquetamiento, producción , marketing, ventas, personal para entrenamiento**

Estos pasos se dan especialmente en un producto comercial.

- **Comprobación con el cliente que el sistema está completamente desplegado y que este es consistente con el criterio de evaluación de la Visión**

Los clientes que contrataron el sistema validan el despliegue y la visión contra el sistema.

- **Mejorar el rendimiento futuro del producto a través de lecciones aprendidas**

Esto incluye el documentar las lecciones aprendidas y mejorar el proceso y el ambiente de desarrollo.

Fase de Transición Iteraciones

La fase de transición puede variar desde ser muy directa a ser extremadamente compleja, dependiendo en el tipo de producto.

Un nuevo release de un producto de escritorio existente puede ser muy simple, requiriendo quizá algunos arreglos menores que pueden ser completados en una sola iteración.

El reemplazo de un sistema de control aéreo puede ser extremadamente complejo, requiriendo varias iteraciones en las cuales características adicionales e integraciones con otros sistemas necesitan ser añadidas. Si existía un sistema anterior, puede que ambos sistemas tengan que correr en paralelo, migración de datos, entrenamiento de usuarios, y ajuste de procesos de negocio, entre otros.

Las actividades realizadas en las iteraciones de la fase de Transición dependen en los objetivos del proyecto. La mayoría de proyectos tienen una relativamente simple transición. Ocasionalmente, nuevas características pueden que tengan que ser añadidas y la iteración se torna similar a una de la fase de Construcción requiriendo trabajo en los requerimientos, análisis y diseño, etc.

Para proyectos muy complejos se puede requerir una modalidad de entregas incrementales del sistema, cada entrega siendo una versión de más alta calidad. Este enfoque es necesario si la

única manera de ajustar el sistema es con retroalimentación de los usuarios que se encuentran operando el sistema en producción.

Para el final de la etapa de Transición, los objetivos de la fase deben haber sido completados y el proyecto debe estar en una posición para ser cerrado. Para algunos proyectos, el final del actual ciclo de vida puede coincidir con el comienzo de otro ciclo de vida conduciendo a la siguiente versión del mismo producto.

Para otros proyectos, el final de la Transición puede coincidir con una completa entrega de los artefactos a una tercera empresa que puede ser responsable de la operación, mantenimiento, y mejoras al sistema entregado.

Un pase por las cuatro fases del RUP es un Ciclo de Desarrollo: al final de la etapa de Transición se ha completado un ciclo de desarrollo. Cada ciclo de desarrollo produce la generación de software. A menos que el producto sea retirado de circulación, este evolucionará en una nueva versión repitiendo la misma secuencia de Inicio, Elaboración, Construcción, y Transición.

2.6 TECNOLOGÍAS OPEN SOURCE

2.6.1 CONCEPTO DE OPEN SOURCE

"El código abierto describe los principios y metodologías que promueven el acceso abierto a la creación y diseño de diversos bienes, productos y recursos. La expresión suele aplicarse al código fuente de software disponible para el público en general con unas restricciones sobre propiedad intelectual muy flexibles o inexistentes. Esto permite a los usuarios crear contenidos de software generados por ellos mismos a través de aportaciones individuales o colaboración." (Wikipedia "open source": http://en.wikipedia.org/wiki/Open_source; 2007-02-15)

•"El código abierto promueve la fiabilidad y calidad del software al permitir las revisiones independientes de los colaboradores y una rápida evolución del código fuente. Para obtener la certificación OSI (Open Source Initiative) el software debe distribuirse bajo una licencia que garantice el derecho a leerlo, redistribuirlo, modificarlo y usarlo libremente" (Opensource.org: <http://opensource.org/search/node>; 2007-02-20)

Open source o código abierto es una alternativa viable al software comercial o también llamado Closed source. El software de código abierto tiene la ventaja y la desventaja de poner al usuario en control de la situación, lo cual implica mayor responsabilidad. En cambio con el software comercial, el usuario se pone a merced del fabricante a cada instancia:

- Cuando no se instala el software.
- Cuando se descubre un defecto o bug.
- Cuando se descubre que el software no hace lo que el usuario esperaba.

- Cuando se descubre que la documentación es demasiado generalizada.

Se puede dar el caso también que el fabricante ha cambiado los términos de licencia y para seguir utilizando el software le toca al usuario seguir pagando una tarifa anual.

Así que el hecho de tener el control sobre el software, y el hecho que algunas aplicaciones de software libre son mejores o equivalentes a su contraparte comercial, hace que nos adentremos en el mundo del Open Source.

En si esta libertad se puede apoyar apreciar de cuatro formas, según Richard Stallman en su trabajo sobre el software libre:

- Libertad para ejecutar el programa en cualquier sitio , con cualquier propósito y para siempre.
- Libertad para estudiarlo y adaptarlo a nuestras necesidades. Esto exige acceso al código fuente.
- Libertad de redistribución, de modo que se nos permita colaborar con vecinos y amigos.
- Libertad para mejorar el programa y publicar las mejoras. También exige el código fuente.

Estas libertades están apoyadas en una licencia la cual también impone ciertas restricciones como el dar reconocimiento a los autores del software en caso de redistribución, incluso puede requerirnos que mejoremos los programas ajenos.

2.6.2 Ambigüedad del término “gratis”

El termino software libre no significa software gratis. La palabra inglesa “free” tiene dos significados: gratis o **libre**, lo que genera una mala interpretación. El software libre puede también ser vendido pero debido a la libertad número 3 que implica la libre distribución, sería imposible hacer dinero, más que obtener una pequeña contribución por grabarlo en un medio físico o realizar el envío del mismo , algo que tampoco se da generalmente por la existencia del internet y su difusión por casi todos los rincones del planeta.

2.6.3 Código Abierto vs. Software Libre

Se podría decir hasta ahora que son términos semejantes, iguales , equivalentes . En verdad , son términos equivalentes pero muy distintos debido a que se enfatiza la disponibilidad del código fuente (Open source) , mas no en términos de la libertad , ya que no se permite realizar modificaciones al código fuente , más se puede incluir programas adicionales o parches , como sucede con las distribuciones de Linux Debian , por ejemplo.

Entonces ahora nos podemos dar cuenta que existen dos organizaciones con ideologías equivalentes pero distintas:

1. Open Source Initiative
2. The Free Software Foundation

2.6.4 OPEN SOURCE INITIATIVE (OSI)

Basándonos en la definición oficial que la Open Source Initiative (OSI)[328] hace en su web podemos dar la siguiente definición de Open Source (Software de Código Abierto):

Código Abierto no solo significa acceso al código fuente. Los términos de distribución del software de código abierto deben cumplir los siguientes criterios

Libre Redistribución.

La licencia no restringirá la venta o cesión del software como componente de una distribución de software que contenga programas de varios códigos distintos. La licencia no requerirá el pago de derechos así como cualquier otra forma de pago por la venta.

Código Fuente

El programa incluirá el código fuente, y deberá permitir la distribución del código fuente como tal así como en forma compilada. En el caso de que el producto distribuido no incluya el código fuente debe haber una forma suficientemente clara de obtenerlo, por un precio razonable no mayor del coste de reproducción, preferentemente a través de la descarga de Internet sin coste alguno. El código fuente debe ser la forma preferida en la que un programador pueda modificar el programa. No se permitirá el código fuente deliberadamente ofuscado, ni tampoco formas intermedias como salidas del preprocesador o traductor.

Trabajo derivado

La licencia debe permitir modificaciones y trabajos derivados, y su distribución en los mismos términos de licencia que el software original.

Integridad del código fuente del autor

La licencia puede restringir la distribución modificada del código fuente solo si la licencia autoriza la distribución de parches o patch files con el código fuente, con la finalidad de modificar el programa en tiempo de compilación. La licencia debe hacer explícito el permiso de distribución de software generado a partir de código fuente modificado. La licencia puede requerir que los trabajos derivados lleven un nombre o número de versión diferente al del software original.

No discriminación contra personas o grupos de personas.

La licencia no puede discriminar a ninguna persona o grupo de personas.

Discriminación contra áreas de trabajo.

La licencia no puede restringir a nadie el uso del programa en un campo específico de trabajo. Por ejemplo, no se puede restringir el uso del programa en un negocio, o para investigación genética.

Distribución de la licencia.

Los derechos vinculados al programa deben ser de aplicación para todos aquellos a los que el programa sea redistribuido, sin necesidad de la ejecución de una nueva licencia para aquellas partes.

La licencia no debe ser específica para un producto

Los derechos vinculados al programa no pueden depender de que el programa sea parte de una distribución de software particular. Si el programa se extrae de esa distribución y se usa o distribuye de acuerdo a los términos de licencia del programa, todas las partes a las que se distribuye el programa deben tener los mismos derechos que los que están licenciados en la distribución de software original.

La licencia no debe restringir otro software.

La licencia no puede plantear restricciones en otro software que es distribuido acompañando al software licenciado. Por ejemplo, la licencia no puede insistir en que todos los restantes programas distribuidos en el mismo medio deben ser de Código Abierto.

La licencia debe ser tecnológicamente neutral

Ninguna previsión de la licencia puede estar dedicada a una tecnología individual o estilo de interfaz.

2.6.5 THE FREE SOFTWARE FOUNDATION (FSF)

Desde otro punto de vista la Free Software Foundation (FSF) define Software Libre de la siguiente manera:

La Free Software Foundation (FSF)[168] manifiesta que:

“El Software Libre es un asunto de libertad, no de precio”.

Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a las cuatro libertades de los usuarios del software que mencionamos al comienzo de este capítulo, pero que revisaremos de nuevo:

- 1. La libertad de usar el programa, con cualquier propósito.**
- 2. La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades. El acceso al código fuente es una condición previa para esto.**
- 3. La libertad de distribuir copias, con lo que puedes ayudar a tu vecino.**
- 4. La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.**

Un programa es Software Libre si los usuarios tienen todas estas libertades. Así pues, se debería tener la libertad de distribuir copias, sea con o sin modificaciones, bien gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar. El ser libre de hacer esto significa (entre otras cosas) que no haya que pedir o pagar permisos. También se debería tener la libertad de hacer modificaciones y utilizarlas de manera privada en el trabajo u ocio de cada cual, sin ni siquiera tener que anunciar que dichas modificaciones existen.

Si se publica cambios, no se tendría por qué avisar a nadie en particular, ni de ninguna manera en particular.

La libertad para usar un programa significa la libertad para cualquier persona u organización de usarlo en cualquier tipo de sistema informático, para cualquier clase de trabajo, y sin tener obligación de comunicárselo al desarrollador o a alguna otra entidad específica.

La libertad de distribuir copias debe incluir tanto las formas binarias o ejecutables del programa como su código fuente, sean versiones modificadas o sin modificar (distribuir programas de modo ejecutable es necesario para que los sistemas operativos libres sean fáciles de instalar).

Está bien si no hay manera de producir un binario o ejecutable de un programa concreto (ya que algunos lenguajes no tienen esta capacidad), pero se debe tener la libertad de distribuir estos formatos si se encontrara o desarrollara la manera de crearlos. Para que las libertades de hacer modificaciones y de publicar versiones mejoradas tengan sentido, debes tener acceso al código fuente del programa. Por lo tanto, la posibilidad de acceder al código fuente es una condición necesaria para el Software Libre.

Para que estas libertades sean reales, deben ser irrevocables mientras no se haga nada incorrecto; si el desarrollador del software tiene el poder de revocar la licencia aunque no se le hayan dado motivos, el software no es libre.

Son aceptables, sin embargo, ciertos tipos de reglas sobre la manera de distribuir Software Libre, mientras no entren en conflicto con las libertades centrales. Por ejemplo, Copyleft es la regla que implica que, cuando se redistribuya el programa, no se pueden agregar restricciones

para denegar a otras personas las libertades centrales. Esta regla no entra en conflicto con estas libertades, más bien las protege.

Software libre no significa no comercial. Un programa libre debe estar disponible para uso comercial, desarrollo y distribución. El desarrollo comercial del Software Libre ha dejado de ser inusual; el Software Comercial Libre es muy importante.

Licencias de Software

Una licencia de software es la autorización o permiso concedido por el titular del derecho de autor, en cualquier forma contractual, al usuario de un programa informático.

La licencia, que puede ser gratuita o excesivamente costosa, precisa los derechos (de uso, modificación o redistribución) concedidos a la persona autorizada y sus límites. Además, puede señalar el plazo de duración, el ámbito de aplicación y todas las demás cláusulas que el titular del derecho de autor establezca. Hay que decir que la licencia de software se puede estudiar desde el punto de vista de los derechos que el autor del software se reserva sobre su obra, o bien desde el punto de vista del usuario final.

2.6.6 Licencias Reconocidas por la FSF y la OSI

A continuación vamos a mostrar un resumen de las licencias aceptadas por las dos principales organizaciones de referencia en cuanto a Software Libre.

Cuadro 3.2: Licencias Open Source y Software Libre

Característica de la licencia vs. Tipo de software	Cero precio	Redistribuible	Uso Ilimitado	Código fuente disponible	Código fuente modificable	Public "Check-ins" al código core	Todos los derivados deben ser libres
Open Source (Linux/GNU style)	X	X	X	X	X	X	X
Open Source (Apache Style)	X	X	X	X	X	X	
Open Source (BSD-Style)	X	X	X	X	X		
Royalty-free libraries	X	X	X	X			
Royalty-free binaries (Freeware)	X	X	X				
Shareware	X	X					
Non-Commercial Use	X	X					
Trial Software	X	X					
Comercial							

Fuente: http://www.iac.com.mx/computo_98/licencias.html

Cuadro 3.3: Licencias aprobadas por la FSF y la OSI

Nombre de Licencia	Aprobado por
CeCILL version 2	fsf
Interbase Public License, Version 1.0	fsf
Historical Permission Notice and Disclaimer	osi
XFree86 1.1 License	fsf
Zend License, Version 2.0	fsf
PHP License, Version 3.01	fsf
Attribution Assurance Licenses	osi
GNU Library or "Lesser" General Public License (LGPL)	osi

Mozilla Public License 1.0 (MPL)	osi
Open Group Test Suite License	osi
Adaptive Public License	osi
OCLC Research Public License 2.0	osi
Modified BSD license	fsf
Sun Industry Standards Source License (SISSL)	osi
GNU Lesser General Public License (LGPL) version 2.1	fsf
Frameworkx License	osi
Academic Free License	osi -fsf
Ricoh Source Code Public License	osi
Python Software Foundation License	osi
Python license	osi
OpenLDAP License, Version 2.7	fsf
LaTeX Project Public License 1.3a	fsf
Expat License	fsf
X.Net License	osi
MITRE Collaborative Virtual Workspace License (CVW License)	osi
LaTeX Project Public License 1.2	fsf
Apache License, Version 1.1	fsf
Zope Public License version 2.0	fsf
PHP License	osi
OpenSSL license	fsf
License of ZLib	fsf
zlib/libpng license	osi
License of Perl 5 and below	fsf
RealNetworks Public Source License V1.0	osi- fsf
Zope Public License version 1	fsf
Cryptix General License	fsf
Original BSD license	fsf
Nokia Open Source License	osi -fsf
License of Vim, Version 6.1 or later	fsf
Apache Software License	osi
GNU General Public License (GPL) version 2	fsf
Fair License	osi
Qt Public License (QPL)	osi
Clarified Artistic License	fsf
IBM Public License	osi
Apple Public Source License	osi-fsf

Open Software License	osi
eCos license version 2.0	fsf
Sybase Open Watcom Public License 1.0	osi
Netizen Open Source License (NOSL),	fsf
W3C License	osi
Jabber Open Source License	osi
Eiffel Forum License	osi
License of Netscape Javascript	fsf
Reciprocal Public License	osi
EU DataGrid Software License	osi -fsf
License of xinetd	fsf
Affero General Public License version 1	fsf
Old OpenLDAP License, Version 2.3	fsf
License of the iMatix Standard Function Library	fsf
New BSD license	osi
Berkeley Database License	fsf
Nethack General Public License	osi
Lucent Public License Version 1.02	osi- fsf
Eclipse Public License	osi- fsf
Zope Public License	osi
University of Illinois/NCSA Open Source License	osi -fsf
Condor Public License	fsf
FreeBSD license	fsf
IBM Public License, Version 1.0	fsf
CUA Office Public License Version 1.0	osi
License of Ruby	fsf
Apache License, 2.0	osi -fsf
Jabber Open Source License, Version 1.0	fsf
Sleepycat License	osi
License of Python 1.6a2 and earlier versions	fsf
Artistic license	osi
Standard ML of New Jersey Copyright License	fsf
Artistic license 2.0	osi -fsf
Lucent Public License (Plan9)	osi
Common Public License 1.0	osi fsf
NASA Open Source Agreement 1.3	osi
GNU Lesser General Public License (LGPL) version 3	fsf
Vovida Software License v. 1.0	osi
Open Software License, all versions through 3.0	fsf

Common Development and Distribution License	osi- fsf
Public Domain	fsf
wxWindows Library License	osi
Educational Community License, Version 2.0	osi
X11 License	fsf
Mozilla Public License 1.1 (MPL)	osi -fsf
Sun Public License	osi- fsf
License of Python 2.0.1, 2.1.1, and newer versions	fsf
GNU General Public License (GPL)	osi
Boost Software License	fsf
Eiffel Forum License V2.0	osi -fsf
Naumen Public License	osi
Entessa Public License	osi
W3C Software Notice and License	fsf
Computer Associates Trusted Open Source License 1.1	osi
Sun Industry Standards Source License 1.0	fsf
GNU General Public License (GPL) version 3	fsf
Motosoto License	osi
MIT license	osi
Phorum License, Version 2.0	fsf
Apache License, Version 1.0	fsf
License of Python 1.6b1 through 2.0 and 2.1	fsf
Q Public License (QPL), Version 1.0	fsf
Netscape Public License (NPL)	fsf
Intel Open Source License	osi -fsf

Fuente: <http://www.asheesh.org/note/software/osi-vs-fsf.html>

2.7 EL SERVIDOR HTTP APACHE

Apache es un servidor web de código libre robusto cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada.

Estos voluntarios se conocen como el Apache Group. Además del Apache Group, cientos de personas han contribuido al proyecto con código, ideas y documentación.

2.7.1 Ventajas

Las ventajas del uso del servidor Apache se pueden resumir en lo siguiente:

Modular

Open Source

Multipataforma

Extensible

Fácil obtención

Apache se encuentra en la versión 2.2.14 en Octubre del 2009, ha sido y sigue siendo el más popular servidor HTTP.

2.8 EL MOTOR DE BASE DE DATOS MYSQL

Mysql es un motor de base de datos utilizado en muchos sitios web. El término velocidad es lo que estaba en la mente de los desarrolladores desde el comienzo, y para eso decidieron incluir menos características que sus competidores más notorios como Oracle, Sybase , etc. A pesar de esto , Mysql tiene las características que necesitan la mayoría de desarrolladores de bases de datos , además es fácil de instalar y la diferencia en precio es enorme.

Mysql es desarrollado, comercializado y mantenido por Mysql AB, la cual es una compañía sueca . Esta compañía brinda dos tipos de licencia:

2.8.1 Licencia Open Source: Mysql está disponible vía licencia GNU GPL (General Public License) sin costo alguno. Cualquier persona que cumpla con los estándares de la GPL puede

usar el software gratis. Si se está utilizando Mysql como motor de base de datos en un sitio web , se puede utilizar Mysql gratis, aún si se está haciendo dinero en el sitio web.

2.8.2 Licencia Comercial: Mysql está disponible con una licencia comercial. Si un desarrollador desea utilizar Mysql , como parte de un nuevo producto de software y desea vender ese producto , el desarrollador necesita obtener una licencia comercial , para la cual la tarifa es bastante razonable.

El encontrar apoyo técnico para Mysql no es un problema. Uno se puede hacer miembro de varios grupos de apoyo en el internet, a través del sitio web de Mysql.

2.8.3 Ventajas

Mysql es una base popular entre los desarrolladores Web del mundo por su velocidad y tamaño compacto. Estos dos elementos la hacen ideal para un sitio web. Además es un producto open source.

A continuación se detalla algunas de sus principales ventajas:

- **Es rápida.** El objetivo principal de los desarrolladores era velocidad. Consecuentemente, el software fue diseñado con eso en mente.
- **Es económica.** Mysql es gratis bajo la licencia GPL open source y la tarifa a pagar por la versión comercial es realmente baja.
- **Es fácil de utilizar:** Se puede construir e interactuar con una base de datos de Mysql utilizando el lenguaje SQL , que es el estándar para comunicarnos con los motores de base de datos.

- **Puede correr en muchos sistemas operativos.** MySQL corre en una variedad de sistemas operativos incluyendo: Windows, Linux , FreeBSD, OS-2 , Irix , entre otros.
- **Soporte técnico ampliamente disponible.** Una gran cantidad de usuarios proveen soporte gratis. Los desarrolladores también participan en listas de correo. También se puede adquirir soporte técnico directamente de MySQL por una módica tarifa.
- **Es segura.** Mysql posee un sistema de autorizaciones que permite todos o algunos privilegios a usuarios específicos. Los password son encriptados.
- **Es multiplataforma.** Mysql se puede instalar en una variedad de sistemas operativos entre los que tenemos:
 - AIX 4.x, 5.x con subprocessos nativos.
 - BSDI 2.x con el paquete MIT-pthreads.
 - BSDI 3.0, 3.1 y 4.x con subprocessos nativos.
 - Digital Unix 4.x con subprocessos nativos
 - FreeBSD 2.x con el paquete MIT-pthreads.
 - FreeBSD 3.x and 4.x con subprocessos nativos.
 - FreeBSD 4.x con LinuxThreads.
 - HP-UX 10.20 con el paquete DCE threads o MIT-pthreads.
 - HP-UX 11.x con subprocessos nativos
 - Linux 2.0+ con LinuxThreads 0.7.1+ o glibc 2.0.7+.
 - Mac OS X.
 - NetBSD 1.3/1.4 Intel y NetBSD 1.3 Alpha (requiere GNU make
 - Novell NetWare 6.0.

- OpenBSD > 2.5 con subprocesos nativos. OpenBSD < 2.5 con el paquete MIT-pthreads.
- OS/2 Warp 3, FixPack 29 y OS/2 Warp 4, FixPack 4.
- SCO OpenServer 5.0.X con una versión del paquete FSU Pthreads recientemente portada.
- SCO UnixWare 7.1.x.
- SCO Openserver 6.0.x.
- SGI Irix 6.x con subprocesos nativos.
- Solaris 2.5 y posteriores con subprocesos nativos en SPARC y x86.
- SunOS 4.x con el paquete MIT-pthreads package.
- Tru64 Unix.
- Windows 9x , Me , NT, 2000, XP , 2003 , Vista.

Fuente: <http://dev.mysql.com/doc/refman/5.0/en/which-os.html>

- **Escalabilidad.** Mysql puede crecer dentro de los siguientes limites:
 - Máximo 64 índices por tabla
 - Cada índice puede incluir hasta 16 columnas
 - Más de 60000 tablas
 - Más de 5.000.000.000.000 registros.

- **Conectividad.** Se puede conectar a la base mediante varios métodos en cualquiera de las plataformas antes descritas. Los métodos principales de conexión a Mysql son:
 - Sockets TCP_IP
 - A través de memoria compartida con la opción `–shared—memory`

- Mediante ODBC (Open Database Connectivity)
 - A través de JMySQL para conexiones con clientes Java que usan JDBC.
-
- **Manejo de concurrencia.** Mysql puede manejar altos niveles de concurrencia sin afectar al funcionamiento de la base ni la integridad de los datos.

 - **Transaccionalidad.** La transaccionalidad propia de las bases comerciales como Oracle , Sql Server , entre otras , se encuentra ya implementada en Mysql. Mediante configuración cambiando el parámetro autocommit a 0 y /o mediante el uso de las sentencias START TRANSACTION y BEGIN.

 - **Asistencia en Línea.** Mysql cuenta con ayuda informativa para todos los comandos o sentencias a través de la opción sentencia–help o sentencia -?.

2.9 PHP

PHP, cuyas siglas responden a un acrónimo recursivo (PHP: hypertext preprocessor), es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como Perl, C y C++. Es rápido, interpretado, orientado a objetos y multiplataforma. Para él se encuentra disponible una multitud de librerías. PHP es un lenguaje ideal tanto para aprender a desarrollar aplicaciones web como para desarrollar aplicaciones web complejas. PHP añade a todo eso la ventaja de que el intérprete de PHP, los diversos módulos y gran cantidad de librerías desarrolladas para PHP son de código libre, con lo que el programador de PHP dispone de un impresionante arsenal de herramientas libres para desarrollar aplicaciones.

PHP suele ser utilizado conjuntamente con Perl, Apache, Mysql o PostgreSQL en sistemas Linux, formando una combinación muy económica (todos los componentes son de código libre), potente y versátil. Tal ha sido la expansión de esta combinación que incluso ha merecido conocerse con un nombre propio LAMP (formado por las iniciales de los diversos productos).

Apache, así como algunos otros servidores web, puede incorporar PHP como un módulo propio del servidor, lo cual permite que las aplicaciones escritas en PHP resulten mucho más rápidas que las aplicaciones CGI habituales.

2.9.1 Cómo funciona

Si solicitamos a nuestro servidor una página PHP, éste envía dicha página al intérprete de PHP que la ejecuta (de hecho, no se trata más que de un programa) y devuelve el resultado (generalmente HTML) al servidor web, el cual, a su vez, se lo enviará al cliente. Imaginemos que tenemos una página PHP con el siguiente contenido:

```
<?php echo "<h1>¡Hola mundo!</h1>";?>
```

Si tenemos este código en un fichero con extensión .php el servidor enviará la página al intérprete de PHP, el cual ejecuta la página y obtiene como resultado:

```
<h1>¡Hola mundo!</h1>
```

El servidor se lo enviará al navegador cliente que ha solicitado la página. El mensaje aparecerá en la pantalla de este último. Veremos que PHP permite mezclar en la misma página HTML y PHP, lo que facilita notablemente el trabajo con éste, pero por otro lado supone un peligro, ya que complica el trabajo en caso de que diseñadores de web y programadores trabajen conjuntamente en las páginas.

Disponemos, en los sistemas en los que esté instalado PHP, de un fichero de configuración global de PHP llamado `php.ini` que nos permitirá configurar algunos parámetros globales. Conviene revisar dicho fichero, pues aunque los valores por defecto suelen ser correctos, puede interesarnos realizar algunos cambios.

2.9.2 Sintaxis

Para empezar a comprender la sintaxis del lenguaje, analizaremos un programa mínimo de PHP:

```
<?php
$MYVAR = "1234";
$myvar = "4321";
echo $MYVAR. "<br>\n";
echo $myvar."<br>\n";
?>
```

La ejecución de este programa (su visualización desde un navegador), dará como resultado:

```
1234<br>
4321<br>
```

El primer punto que debemos destacar es que los bloques de código de PHP están delimitados en HTML con `<?php` y `?>`. Podemos, por tanto, escribir una página HTML e incluir en ella diversos bloques de instrucciones PHP:

```
<HTML>

<HEAD>

<TITLE>Título del documento</TITLE>

</HEAD>

<BODY>

<h1>Cabecera H1</h1>

<?php echo "Hola" ?>

<h1>Cabecera H1 segunda</h1>

<?php

$MYVAR = 1234;

$myvar = 4321;

echo $MYVAR. "<br>";

echo $myvar."<br>";

// Este programa presenta en pantalla unos números

?>

</BODY>

</HTML>
```

El siguiente punto que conviene destacar es que los nombres de variables se distinguen en que siempre deben empezar con \$, y que igual que en C/C++, son case-sensitive, es decir, diferencian mayúsculas y minúsculas. Fijémonos también en que para concatenar texto (las variables y `
`) utilizamos el carácter punto "." y además, en que todas las sentencias

terminan con “;”. Asimismo conviene observar que las variables, a pesar de ser numéricas, se pueden concatenar con un texto (“
”). En este caso el intérprete convierte el valor numérico de la variable en texto para poder realizar la concatenación.

Observemos el comentario dentro del código. Este comentario no afectará en modo alguno al programa ni será enviado al navegador del cliente (de hecho, el navegador cliente nunca recibirá código PHP). Para introducir comentarios en nuestro código, disponemos de dos opciones:

```
// Comentario de una sola línea
```

```
/* Esto es un comentario de varias líneas.
```

```
Para ello usamos este otro marcador
```

```
de inicio y final de comentario */
```

2.9.3 Variables

PHP no precisa que declaremos por adelantado la variable que vamos a usar ni el tipo de ésta. PHP declarará la variable y le asignará el tipo de datos correcto en el momento en que la usemos por primera vez:

```
<?php $cadena = “Hola Mundo”;
```

```
$numero = 100;
```

```
$decimal = 8.5;
```

```
?>
```

Como podemos observar, las tres variables fueron definidas en el momento de asignarles valor y no tuvimos que definir tipos. En PHP las variables pueden tener, básicamente, dos ámbitos:

- Global, en el que serán accesibles desde todo el código y
- Local, en el que sólo serán accesibles desde la función en la que las creamos.

Para asignar a una variable un ámbito global bastará con declararla (en este caso, sí que hace falta una declaración de variable y usar la palabra reservada `global` en la declaración:

```
<?php
global $test;
?>
```

Las variables que no calificamos como globales, pero que sean de finidas fuera de cualquier función, tendrán como ámbito el global. Bastará con definir una variable dentro de una función. En ese caso su ámbito quedará restringido a la función donde la declaremos.

```
<?php
global $variable; // Variable global

$a=1; // Variable global implícita

function suma()
{
    $b=1; // b es una variable local

    $res=$a+$b; // res es una variable local
}
?>
```

Podemos ver que tanto `a` como `variable` son variables globales, mientras que `b` y `res` son variables locales. Además, disponemos en PHP de variables de vectores o arrays . Éstas son variables que pueden contener listas de elementos, a los que accederemos por medio de un índice.

```
<?php
$mares = array(); //con array() declaramos un vector
$mares[0]= "Mediterráneo";
$mares[1] = "Aral";
$mares[2] = "Muerto";
?>
```

Como podemos ver, hemos declarado la variable `mares` con una llamada a `array()`. Esto indica a PHP que dicha variable es un vector de elementos. Para acceder a los elementos individuales del vector, debemos utilizar el nombre del vector e indicar la posición del elemento al que queremos acceder entre corchetes. En PHP los vectores empiezan a numerarse en 0. Además de vectores con índices numéricos, PHP soporta vectores cuyos índices sean cadenas de texto . Estos se llaman vectores o arreglos asociativos

```
<?php
$montañas = array(); //con array() declaramos un vector
$montañas["Everest"]="Himalaya";
$montañas["Fitz Roy"] = "Andes";
$montañas["Montblanc"] = "Alpes";
echo $montañas["Everest"]; // Imprimirá Himalaya
?>
```


2.9.4 Funciones de cadenas

PHP provee de un conjunto de funciones muy interesantes para el trabajo con cadenas de texto. Algunas de las más destacadas son:

strlen Devuelve la longitud de una cadena.
explode Divide una cadena en función de un carácter separador, y devuelve un vector con cada uno de los trozos de la cadena.
implode Actúa al revés que explode, uniendo diversas cadenas de un vector con un carácter de unión.
strcmp Compara dos cadenas a nivel binario.
strtolower Convierte una cadena a minúsculas.
strtoupper Convierte una cadena a mayúsculas.
chop Elimina el último carácter de una cadena, útil para eliminar saltos de línea o espacios finales superfluos.
strpos Busca dentro de una cadena otra cadena especificada y devuelve su posición.
str_replace Reemplaza en una cadena una aparición de una subcadena por otra subcadena. Esto nos sirve cuando queremos remover caracteres que no queremos almacenar en nuestra base de datos o mostrar.

Podemos ver el funcionamiento de algunas de estas funciones en el siguiente ejemplo:

```
<?php
$cadena1 = "hola";
$cadena2 = "pera,manzana,fresa";
$longitud = str_len($cadena1); //longitud=4
$partes = explode(",",$cadena2);
//genera el array $partes con $partes[0]="pera",
//$partes[1]="manzana"; y $partes[2]="fresa";
$chop = chop($cadena); // chop elimina la "a"
$cadena3 = str_replace(",","",$otracadena);
```

```
//$cadena3 contiene: pera-manzana-fresa  
//Cambiamos las , por -  
?>
```

2.9.5 Acceso a ficheros

PHP proporciona un amplio repertorio de métodos para el acceso a ficheros. Vamos a mostrar la más práctica y simple, muy adecuada si los ficheros a los que accederemos son pequeños.

El código que comentaremos es el siguiente:

```
<?php  
$fichero = file("entrada.txt");  
$numlin = count($fichero);  
for($i=0; $i < $numlin; $i++)  
{  
echo $fichero[$i];  
}  
?>
```

En este ejemplo leemos un archivo que tiene por nombre entrada.txt y lo mostramos como salida. El primer paso que seguimos es declarar la variable fichero, lo que nos genera un vector en el que PHP colocará todas las líneas del archivo. Para ello usaremos la función de librería file. El siguiente paso consiste en averiguar cuántos elementos contiene fichero. Para ello usamos la función count, que nos devuelve el tamaño de un vector, en este caso el vector que hemos generado al leer el fichero. Finalmente podemos escribir un bucle que recorrerá el vector tratando cada una de las líneas del archivo.

PHP proporciona muchas funciones más de tratamiento de ficheros. Disponemos, por ejemplo, de la función **fopen**, que permite abrir ficheros o recursos sin leerlos íntegramente en memoria y que es capaz de abrir ficheros de la siguiente forma:

```
<?php
$recurso = fopen ("entrada.txt", "r");
$recurso = fopen ("salida.gif", "wb");
$recurso = fopen ("http://www.uoc.edu/", "r");
$recurso = fopen (
"ftp://usuario:password@uoc.edu/salida.txt", "w");
?>
```

Ahí podemos ver cómo abrimos un fichero para lectura (“r”), para escritura binaria (“wb”), una página web para leerla como si se tratara de un fichero y un fichero vía FTP para escribirlo, respectivamente.

2.9.6 Acceso a bases de datos

PHP proporciona métodos para acceder a un gran número de sistemas de bases de datos (mySQL, PostgreSQL, Oracle, etc.).

Esta funcionalidad es imprescindible para el desarrollo de aplicaciones web complejas.

2.10 ACTIVE RECORD

Active Record es un patrón de acceso y manipulación de las bases de datos que se ha hecho popular últimamente.

Active Record conecta objetos del negocio con tablas del esquema de base de datos para así crear un modelo de dominio persistente donde los datos y la lógica son presentados al mismo tiempo implementando el famoso modelo ORM , como fue descrito por Martin Fowler:

“Un objeto que se relaciona con una fila en una base de datos o vista, encapsula el acceso a la base de datos y añade lógica del dominio del negocio a tales datos”.

La contribución que hace Active Record es brindar una abstracción de las sentencias SQL, lo cual ayuda enormemente al mantenimiento del sistema. Es así que Active Record posee métodos que encapsulan las sentencias SELECT, INSERT, UPDATE, DELETE, etc , por lo que ya no debemos preocuparnos de la sintaxis de estas ya que se realizan “tras bastidores” .

Ejemplo:

Tenemos la clase Usuarios

```
$usuario= new Usuarios();
```

```
$usuario->nombre="fjacome";
```

```
$usuario->password="dfd253uhtewq";
```

```
$usuario->rol="Administrador";
```

```
$usuario.save();
```

La última sentencia equivale a realizar un insert

```
Insert into Usuarios(nombre,password,rol) values (“fjacome”, “ dfd253uhtewq”, “  
Administrador”);
```

Además se puede consultar a la base de esta manera

```
$usuario.findFirst(“nombre”, “fjacome”);
```

La operación de update se realiza de esta manera:

```
$usuario.save();
```

El objeto usuario persiste y como ya existe en la tabla, se realiza un UPDATE internamente.

Active Record facilita enormemente el desarrollo y permite realizar las aplicaciones en forma ordenada, más rápida y eficiente.

CAPITULO III

AJAX

3.1 Definición

“AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.¹

De la definición anterior podemos destacar los siguientes puntos:

- Ajax es asíncrono
- Envuelve Javascript y XML
- Sirve para desarrollar aplicaciones RIA

¹ Fuente: <http://es.wikipedia.org/wiki/AJAX>

- Mejor Interactividad con el usuario

En Febrero del 2005 , Jesse James Garret de la empresa Adaptive Path , publicó un artículo en línea llamado “Ajax: Un nuevo enfoque hacia las aplicaciones web” , el cual todavía se encuentra en la dirección:

www.adaptivepath.com/publications/essays/archives/000385.php

Aquí Garret explicó como él pensaba que las aplicaciones web estaban cerrando la distancia entre el Web y las aplicaciones tradicionales de escritorio. El citó nuevas tecnologías y varios de los proyectos de Google como ejemplos de cómo interacciones típicas de aplicaciones de escritorio estaban siendo usadas en el Web. El concluyó con dos sentencias que iniciaron todo: “Note que Google Suggest y Google Maps son dos ejemplos de un nuevo enfoque a las aplicaciones Web que nosotros en Adaptive Path hemos llamado Ajax. El Nombre es compuesto de Asynchronous JavaScript + XML y representa un giro fundamental hacia lo que es posible de realizar en el Web”

Desde ese día, muchos artículos sobre Ajax, ejemplo de código y debates comenzaron a aparecer en todo el Web. Pero esto no se dio de la noche a la mañana ya que la evolución de varias tecnologías ocurrió primero.

3.2 La Evolución del Web

Cuando Tim Berners Lee preparó la primera propuesta para la World Wide Web en el año 1990, la idea consistía en crear una red de información interconectada usando hipertexto y URIs. La habilidad de unir documentos dispersos que se encontraban en todo el mundo tenía mucho potencial para propósitos educativos ya que las personas serían capaces de acceder a

este material casi de manera instantánea. La primera versión de HTML (HyperText Markup Language) tenía únicamente texto e información estática. A medida que el Web crecía, los negocios vieron un potencial para distribuir información acerca de sus productos y servicios a las masas. La siguiente generación del Web experimentó una mayor habilidad de formatear y desplegar información como HTML, lo que vino a satisfacer las necesidades de publicidad de los negocios interesados. Pero en un lugar, una pequeña compañía llamada Netscape, pronto estaría lista para llevar la revolución del Web hacia el futuro a paso más rápido.

3.2.1 JavaScript

Netscape Navigator fue el primer navegador de la Web exitoso. Sin embargo, Netscape fue a menudo criticado por implementar nuevas tecnologías y extensiones a tecnologías existentes antes de que los estándares estén listos. Una de esas tecnologías fue JavaScript, originalmente llamado LiveScript, fue creado por Brendan Eich de Netscape y lo incluyó en la versión 2.0 del navegador lanzada en 1995. Por primera vez, los desarrolladores tuvieron la capacidad de experimentar como una página web podía interactuar con el usuario. En lugar de hacer viajes constantes al servidor para tareas simples como validación de datos, JavaScript hizo posible el transferir estas tareas al navegador, osea al cliente. Esto era muy importante para los usuarios de Internet que al momento tenían conexiones dial-up a un máximo de 28.8 Kbps que debían soportar tiempos de respuesta enormes. El minimizar ese tiempo de espera mediante menos viajes al servidor fué el primer paso hacia Ajax.

3.2.2 Frames o Marcos

La versión 4.0 de HTML introdujo oficialmente el concepto de los Marcos. La idea de que el despliegue de una página web podría ser dividido en varios documentos fue calificada como **radical y controversial** al mismo tiempo que Netscape escogió implementar esa característica aún antes que HTML 4.0 fuera completado en el navegador Netscape Navigator 2.0 , el cual llegó a ser el primero en soportar marcos y JavaScript juntos. Esto resultó ser un paso enorme en la evolución de Ajax.

Cuando la guerra de navegadores empezó a finales de los noventa entre Microsoft y Netscape, ambos JavaScript y los marcos fueron formalizados. Nuevas mejoras se añadió a ambos gracias al esfuerzo de desarrolladores creativos que empezaron a experimentar con ambos. Debido a que un marco representaba una petición por separado al servidor, la habilidad de controlar un marco y sus contenidos con JavaScript abrió la puerta a muchas posibilidades emocionantes.

3.2.3 La técnica del marco escondido

A medida que los desarrolladores empezaron a comprender como manipular marcos, una nueva técnica emergió para facilitar la comunicación entre cliente y servidor. La técnica del marco escondido consistía en establecer un set de marcos donde uno de los marcos era seteado a 0 pixeles de alto y 0 pixeles de ancho. El propósito de esta técnica era el iniciar una comunicación con el servidor. El marco escondido u oculto contendría una forma HTML en la que algunos campos específicos serían llenados dinámicamente por JavaScript y retransmitidos al servidor. Cuando el marco retornaba, este llamaba a otra función JavaScript para notificar al marco original que los datos habían sido retornados. Esta técnica representaba

el primer modelo asíncrono para aplicaciones Web. Pero también otro avance tecnológico estaba por llegar.

3.2.4 HTML Dinámico y el DOM

Era el año 1996, el Web era todavía un mundo estático. Aunque JavaScript y la técnica del marco escondido ponían vida a la Web y la interacción con el usuario, no había manera de cambiar el estado de una página sin recargarla. Entonces vino Internet Explorer 4.0. A este punto, Internet Explorer había alcanzado al líder, que era Netscape Navigator y mejoró un aspecto importante con la introducción del DHTML o HTML Dinámico. Aunque este estaba todavía en fase de desarrollo, DHTML significó un paso importantísimo ya permitía hacer lo que no se podía, alterar cualquier parte de una página web sin tener que recargar toda la página.

Después apareció CSS, el cual puso más vigor al desarrollo con DHTML a pesar de las discrepancias entre Microsoft y Netscape. La comunidad de desarrolladores estaba muy contenta ya que ahora se podía combinar DHTML con la técnica del marco escondido para recargar cualquier parte de una página con llamadas asíncronas al servidor. Desafortunadamente, DHTML nunca fue incluido en los estándares debido a la fuerte influencia de Microsoft que introdujo el DOM (Document Object Model) que a diferencia de DHTML, que solo tenía como propósito el modificar secciones de una página, tenía un propósito más ambicioso que era el de proveer una estructura para la página Web entera. La modificación de dicha estructura permitiría modificaciones tipo DHTML a la página. Esto era el siguiente paso hacia adelante para Ajax.

3.2.5 Iframes

Aunque la técnica del marco escondido llegó a ser increíblemente popular, esta tenía una desventaja: Uno debía planificar con anticipación y escribir un frameset previo al uso de marcos escondidos u ocultos. Cuando el elemento iframe fué presentado como una parte del HTML 4.0 en 1997, representó otro paso significativo hacia la evolución del Web y Ajax.

En lugar de definir framesets , los desarrolladores podían colocar iframes en cualquier parte de la pagina. Esto permitía a los desarrolladores olvidarse de framesets y simplemente colocar iframes en cualquier parte de una página (a través del uso de CSS) para habilitar la comunicación cliente-servidor. Cuando el DOM finalmente se implementó en Internet Explorer 4 y Netscape 6, este ofrecía la posibilidad de dinámicamente crear iframes , lo que significaba que una función JavaScript podía ser utilizado para crear iframes, hacer una petición , y obtener una respuesta , todo esto sin incluir HTML adicional en una página. Esto condujo a la siguiente generación de la técnica del marco escondido: La técnica del Iframe escondido.

3.2.6 XMLHttp

Los desarrolladores de Microsoft debieron haberse dado cuenta de la popularidad del marco escondido y la más reciente iframe escondida, que decidieron proveer a los desarrolladores con una mejor herramienta para la interacción cliente-servidor.

Esa herramienta vino en la forma de un objeto Activex llamado XMLHttp presentado en el año 2001. Una de las extensiones de Microsoft para JavaScript permitía la creación de controles ActiveX. Cuando Microsoft comenzó a soportar XML a través de una librería

llamada MSXML, el objeto XMLHttpRequest fue introducido. Aunque llevaba en su nombre XML , este objeto no era sino otra manera de manipular datos XML.

Era como una petición HTTP que podía ser controlada por JavaScript , es así que los desarrolladores tenían acceso a los códigos de estado y cabeceras , así como los datos retornados por el servidor. Esos datos retornados por el servidor podían estar estructurados como XML, objetos JavaScript serializados o datos en cualquier formato escogido por el desarrollador. En lugar de usar marcos escondidos o iframes. Ahora era posible acceder al servidor programáticamente usando JavaScript puro. El objeto XMLHttpRequest llegó a ser tremendamente popular para los desarrolladores que utilizaban Internet Explorer.

Con la popularidad ya marcada del objeto XMLHttpRequest , los desarrolladores del proyecto Open source Mozilla comenzaron su propio desarrollo de un objeto XMLHttpRequest. En lugar de permitir acceso a ActiveX , los desarrolladores de Mozilla replicaron los métodos más importantes del objeto y sus propiedades en un objeto XMLHttpRequest.

Con ambos navegadores implementando alguna forma de XMLHttpRequest , el desarrollo de interfaces tipo Ajax forzó a otros navegadores nuevos como Opera y Safari a soportar alguna forma del objeto (ambos decidieron seguir la ruta de Mozilla).

3.2.7 Ajax propiamente dicho

A pesar de las frecuentes preguntas formuladas sobre Ajax en foros y blogs, todavía existe algo de confusión con respecto a lo que Ajax realmente es. Ajax es nada más que un enfoque en la interacción con el usuario. Este enfoque envuelve transmitir solo una pequeña porción de información hacia y desde el servidor con el objetivo de dar al usuario la más responsiva

experiencia posible. En lugar del modelo tradicional de las aplicaciones Web donde el navegador es el responsable de iniciar las peticiones hacia y procesar las peticiones desde el servidor web, el modelo Ajax provee una capa intermedia , que es lo que Garret llama un Motor Ajax, para manejar esta comunicación. Un Motor Ajax es realmente un objeto JavaScript o función que es llamada siempre y cuando se necesite pedir información del servidor. En vez utilizar la tradicional práctica de proveer un hipervínculo hacia otro recurso, como por ejemplo otra página web, cada hipervínculo hace una llamada al Motor Ajax , el cual ejecuta la petición.

Como se dijo anteriormente, la petición se ejecuta asíncronamente , lo que significa que la ejecución no espera una respuesta para continuar.

El servidor, es configurado para retornar datos que el Motor Ajax puede utilizar. Estos datos pueden ser texto simple , XML , JSON u otro formato de datos que se pueda necesitar. El único requerimiento es que el Motor Ajax pueda comprender e interpretar los datos. Cuando el motor Ajax recibe la respuesta del servidor, este entra en acción, pasando los datos y realizando cambios a la interface basados en la información que fue provista .

Debido a que este proceso envuelve una menor transferencia de datos que una aplicación web tradicional, las actualizaciones a la interfaz del usuario son más rápidas , y el usuario puede realizar su trabajo más rápidamente.

3.2.8 Ajax y JSON

Ajax obtuvo parte de su nombre por su interacción inicial con XML. Pero poco a poco la comunidad de desarrolladores se fue dando cuenta que se debía agilizar el transporte de datos con algo diferente y más optimo que XML. Es así que se llegó a crear JSON que significa

Javascript Object Notation. JSON permite el transporte de datos desde y hacia el cliente liberando a las aplicaciones de las etiquetas y encabezados propios del XML. Otra ventaja añadida es que JSON se traduce directamente a objetos de javascript , lo cual mejora los tiempos de respuesta y procesamiento. Esta es la nueva tendencia que explicaremos más adelante.

3.3 Principios del Ajax para la usabilidad

Como un nuevo modelo de aplicación. Ajax todavía está en sus primeras etapas, sin embargo, varios desarrolladores web han tomado este nuevo desarrollo como un desafío. El desafío consiste en definir que hace a una buena aplicación ajax versus que hace a una mala. Michael Mahernoff , un experto en desarrollo de software y usabilidad identificó varios aspectos claves que se deben tener muy en cuenta a la hora de desarrollar con esta tecnología.

3.3.1 Mínimo trafico

Las aplicaciones Ajax deben enviar y recibir la menor cantidad de información como sea posible desde y hacia el servidor. En otras palabras, Ajax puede minimizar la cantidad de tráfico entre el cliente y el servidor. Este aspecto añade a la robustez de la aplicación.

3.3.2 Evitar las sorpresas

Las aplicaciones Ajax típicamente introducen modelos diferentes de interacción con el usuario que las aplicaciones web tradicionales. A diferencia del estándar clic y espera , algunas

aplicaciones Ajax ofrecen otras opciones como son drag and drop o doble clic. No importa que modelo se utilice mientras se mantenga la consistencia para que el usuario sepa que hacer.

3.3.3 Seguir las Convenciones Establecidas

No se debe perder el tiempo tratando de inventar modelos nuevos de interacción con el usuario con los que este no sea familiar. Se debe tener en cuenta las aplicaciones ya existentes y tomar ideas de ellas. Los usuarios probablemente ya las han utilizado y la aplicación que desarrollemos no será extraña en términos de usabilidad , además la curva de aprendizaje será mínima para estos usuarios.

3.3.4 Evitar las distracciones

Como una regla general debemos evitar elementos innecesarios que distraen al usuario tales como animaciones repetitivas, secciones que se iluminan y se apagan, etc. Esto lo único que consigue es apartar al usuario del objetivo que trata de cumplir.

3.3.5 Accesibilidad

Consideremos quienes van a ser nuestros usuarios primarios y secundarios y como ellos accederán a nuestra aplicación Ajax. Tratemos de incluir al máximo número de usuarios para lo cual el software que estos usen o requieran no sea un limitante.

3.3.6 Evitar descargar la página entera

Todas las comunicaciones después del acceso inicial al sitio deben ser manejadas por el Motor de Ajax. No echemos a perder la experiencia del usuario al descargar cantidades pequeñas de datos en un lugar para después recargar la página entera.

3.3.7 El usuario es primero

Como una buena práctica se debe diseñar la aplicación Ajax con el usuario en mente antes que nada. Tratemos de satisfacer los casos de uso primeramente antes que pensar en efectos , ni como se verá el sitio ,con que colores , etc.

La usabilidad es el patrón común en las anteriores reglas , y eso es comprensible ya que el objetivo de Ajax es el mejorar la experiencia del usuario.

Al adherirnos a estos principios podemos estar seguros que nuestra aplicación Ajax será todo un éxito.

3.4 Patrones de diseño para aplicaciones AJAX

Los patrones de diseño describen técnicas de programación para resolver problemas comunes. Dado que la programación ha estado presente en el medio por varias décadas , puede darse que muchos de los problemas que se dan todos los días ya hayan sido resueltos por alguien más. Desde mediados de los noventas, mucha atención ha estado siendo puesta en los patrones de diseño como una manera de acortar los tiempos de desarrollo.

Aunque el término Ajax ha estado en el medio desde el 2005, las técnicas que Ajax describe han sido utilizadas desde finales de la década de los noventa, dando como resultado la creación de varios patrones para resolver problemas específicos. Ya hemos visto algunos de estos patrones en acción, como la técnica del marco escondido, las llamadas asíncronas mediante el uso del objeto XMLHttpRequest. Estos son patrones de comunicación entre el cliente y el servidor usando JavaScript. Pero existen más patrones de diseño que describiremos más adelante.

El autor y programador Michael Mahemoff fue el primero en documentar los patrones de diseño de Ajax en su sitio web www.ajaxpatterns.org. Los patrones descritos más adelante son una mezcla de la documentación de Mahemoff y otros autores.

Nótese sin embargo que cualquiera de estos patrones nunca puede ser oficial ni ser la última palabra, solo pueden ser aceptados como soluciones que, volvemos a insistir, han resuelto problemas anteriormente.

3.4.1 Patrones de Control de Comunicación

Como ya sabemos , para comunicarnos con el servidor utilizamos JavaScript, pero la pregunta aquí es :Cuál es la mejor manera de iniciar y continuar realizando peticiones al servidor?. En algunos casos, puede ser mejor realizar una carga preliminar para tener el contenido inmediatamente disponible. En otros casos , talvés se desea enviar los datos en intervalos de tiempo. Quizá todo el contenido no deba ser descargado de una sola sino en una secuencia en particular. Ajax permite controlar la comunicación entre el cliente y el servidor para lograr el comportamiento deseado.

3.4.2 El Fetch Predictivo

En una solución web tradicional, la aplicación no tiene ni idea de lo que viene a continuación. Una página es presentada con un número de enlaces, cada uno que lleva a una parte diferente del sitio. Esto puede ser llamado un “Fetch por demanda” , donde el usuario , a través de sus acciones, le dice al servidor exactamente que datos deben ser recuperados. Aunque este paradigma ha definido al Web desde su nacimiento, tiene un lado malo de forzar un modelo de iniciar y parar refiriéndonos a la interacción del usuario. Con la ayuda de Ajax , esto está cambiando.

El patrón de Fetch predictivo es una idea relativamente simple pero algo compleja de implementar. Aquí la aplicación Ajax adivina lo que el usuario hará a continuación y recupera los datos apropiados. En un mundo perfecto , sería maravilloso siempre saber los que el usuario va a hacer y así asegurarnos que los datos correctos están listos para su uso , pero en la realidad, determinar la acción futura del usuario es solo un juego de adivinanzas. Es por eso que se debe diseñar teniendo en cuenta los diferentes escenarios de usabilidad y tener una respuesta para cada uno de ellos. Existen casos de uso sencillos donde el predecir la próxima acción del usuario es fácil. Supongamos que se está leyendo un artículo en línea que está separado en tres páginas. Es lógico asumir que si se está interesado en leer la primera pagina, también se está interesado en leer la segunda y por ende la tercera. Si la primera página ha sido cargada por unos segundos, es probablemente seguro que se deba descargar la segunda página en el background. Así mismo, si la segunda página ha sido cargada por unos pocos segundos, es lógico asumir que el lector continuará con la tercera página. Esto hace que el usuario al pasar a la siguiente página la pueda visualizar inmediatamente después de hacer clic en siguiente.

Otro caso de uso simple se da durante la realización de un email. La mayoría de las veces, escribiremos un mail a alguien que conocemos, así que es lógico asumir que la persona ya está en el libro de contactos. Para ayudar al usuario, debemos pre cargar el libro de contactos y ofrecer sugerencias.

Es así que la clave en estos casos es la frase: “es lógico asumir que ...” . Al realizar estas acciones de carga, es más probable que si hayamos adivinado la acción siguiente a ejecutarse por parte del usuario.

3.4.3 Aceleración del Envío (Submission Throttling)

El Fetch predictivo es un patrón para recuperar datos del servidor, el otro lado de la moneda es el enviar datos al servidor. Debido a que se quiere evitar recargar la página, la pregunta acerca de cuándo enviar los datos del usuario es muy importante. En las aplicaciones web tradicionales, cada clic realiza un viaje al servidor para que el servidor este informado de lo que el cliente está haciendo. En el modelo Ajax , el usuario interactúa con el sitio o las aplicaciones sin tener que generar peticiones adicionales con cada clic.

Una solución sería enviar los datos de vuelta al servidor cada vez que una acción del usuario ocurre, similar a lo que se hace en una solución web tradicional. Es así que cuando el usuario digita una letra, esa letra es enviada al servidor inmediatamente. El proceso es luego repetido por cada letra de la palabra digitada. El problema con este método es que puede crear un alto número de peticiones en un periodo corto de tiempo, lo cual puede que no cause problemas con el servidor pero puede hacer que el interfaz se ponga lenta mientras las peticiones son realizadas y procesadas.

Este patrón es una alternativa al problema arriba descrito , ya que se almacena en un buffer los datos a ser enviados al servidor para que estos datos sean luego enviados al servidor en intervalos de tiempo. Google Suggest tiene una implementación de esto que es brillante. No envía una petición después que cada carácter es digitado. En vez de eso , espera un tiempo , pasado este lapso envía una petición con el contenido de la caja de texto al servidor.

3.4.4 Envío Explícito

Es una alternativa a la Aceleración de envío. En ambos queremos asegurar que la aplicación responda rápidamente pero si lo hacemos para todos los eventos de validación e ingreso de datos en el cliente se volverá lenta y torpe. Así que lo que hacemos es forzar al usuario a que solicite uno o más eventos Ajax. Por ejemplo, si queremos realizar validaciones , cálculos , etc , solo incluimos un botón enviar al final de la forma para que este sea el que inicie la interacción con el servidor.

Otro ejemplo notorio es el de las aplicaciones de chat, en donde se requiere que el usuario presione el botón enviar para enviar sus datos al servidor.

3.4.5 Refresco Periódico

Este patrón describe el proceso de chequear nueva información del servidor en intervalos específicos. Este enfoque requiere que el navegador de internet mantenga un registro de cuando otra petición al servidor debe tomar lugar.

Este patrón es utilizado por los siguientes sitios web de renombre :

ESPN

Las páginas con los resultados se actualizan periódicamente

GMAIL.

Usa el Refresco Periódico para notificar a los usuarios cuando haya nuevo mail recibido.

Mientras el usuario está leyendo el mail o ejecutando otras operaciones

XHTML Live Chat

Utiliza el refresco periódico para implementar una sala de chat usando tecnologías web simples. El texto de la sala de chat es actualizado automáticamente cada pocos segundos ya que chequea el servidor para así encontrar nueva información. Si hay un mensaje nuevo , la página es actualizada para reflejar los cambios.

3.4.6 Descarga Multi-Etapas

Uno de los problemas más grandes en la web ha sido la velocidad en que las páginas se descargan. Cuando se utilizaba los modem de 56 Kbps, los diseñadores web estaban más alerta de cuanto pesaban sus páginas. Con la popularidad de servicios de banda ancha de Internet muchos sitios se han mejorado con imágenes, sonido, video, etc.

Este enfoque, aunque le da al usuario más información, también conduce a tener bajos tiempos de respuesta ya que todo es bajado en orden aleatorio. Afortunadamente, existe una solución Ajax para este problema.

Se carga la más básica funcionalidad inicialmente. Una vez completada la operación, la página empieza a bajar otros componentes que deberían aparecer en la página. Si el usuario abandona la página antes de que todos los componentes sean descargados, no tiene consecuencia. Si el usuario permanece en la página por un periodo extendido de tiempo, la funcionalidad extra es cargada en el background y estará disponible cuando el usuario la necesite. La mayor ventaja radica en que nosotros como desarrolladores decidimos que se descarga y en qué punto en el

tiempo. Este es un patrón de Ajax relativamente nuevo que ha sido popularizado por Microsoft.

3.5 JSON

Hasta hoy hemos revisado Ajax como un conjunto de tecnologías para alcanzar una mejor experiencia del usuario en sus interacciones con aplicaciones Web. Ajax es la unión de JavaScript Asíncrono con XML, pero este concepto del uso de XML aunque ha funcionado relativamente bien, tiene también sus detractores que se basan en argumentos válidos y por lo tanto decidieron divisar una nueva manera de evitar los males de XML y tomar sus virtudes.

Es así que emergió JSON, cuyas siglas en inglés significan Notación de Objetos JavaScript.

Uno de los argumentos que se tiene en contra de XML es el llamado verbose, lo cual significa que se desperdician valiosos bytes de espacio en etiquetas y palabras que no se utilizan realmente. Aunque nuevas formas de comprimir el XML se han creado, hasta una forma de XML binario, estas no han pegado realmente debido a que no tienen una compatibilidad estandarizada con las versiones principales de XML. Es por eso que Douglas Crockford, un ingeniero de software de muchos años, propuso un nuevo formato de datos construido en JavaScript llamado JSON.

Hasta ese punto, se usaba texto simple o XML para transmitir datos hacia y desde el servidor. En esta sección vamos analizar el uso de JSON en la comunicación de Ajax y ya habiendo pesado los beneficios de esta notación, se ha decidido utilizar JSON como formato de comunicación de datos entre el cliente y el servidor del presente proyecto de tesis.

Empezaremos dando una pequeña descripción de lo que es JSON. JSON es un formato de datos liviano basado en un subconjunto de la sintaxis de JavaScript, que incluye vectores y objetos. Debido a que utiliza sintaxis de JavaScript, este puede ser incluido dentro de archivos JavaScript y accedido sin el extra parsing que se hacía con XML. Pero antes de poder utilizar JSON es importante comprender la sintaxis específica para los vectores y objetos.

3.5.1 Vectores

La notación para vectores es generalmente dos corchetes para encerrar a una lista de valores JavaScript delimitados por comas, por ejemplo:

```
var vNombres ["Carlos","Lorena","Ramona"];
```

Se puede acceder a las variables de esta manera:

```
Alert(vNombres[0]); //Nos despliega Carlos
```

```
Alert(vNombres[1]); //Nos despliega Lorena
```

```
Alert(vNombres[2]); //Nos despliega Ramona
```

Debido a que los vectores no corresponden a un tipo de datos específico, pueden ser utilizados para almacenar cualquier tipo de variables de diferentes tipos de datos.

```
var vValores = ["string", 24, true, null];
```

El anterior vector contiene una cadena, un número entero, un booleano y un nulo, lo que es perfectamente legal en JavaScript.

3.5.2 Objetos Javascript

Los objetos JavaScript son utilizados para almacenar información en el esquema nombre-valor. Un objeto de este tipo es definido por dos llaves, dentro de estas puede ser colocado cualquier número de pares nombre-valor, definidos por una cadena, dos puntos y el valor.

Cada uno de estos pares deben ser seguidos de una coma, excepto por el último elemento del objeto. Por ejemplo

```
var oCarro = {  
    "color" : "rojo",  
    "puertas" : 4,  
    "pagado" : true  
};
```

Este código crea un objeto con tres propiedades, las cuales pueden ser accedidas mediante el operador punto.

```
alert(oCarro.color); //despliega "rojo"  
alert(oCarro.puertas); //despliega "4"  
alert(oCar.pagado); //despliega "true"
```

También se puede utilizar la siguiente notación con corchetes para pasar en el nombre de la propiedad el valor como una cadena.

```
alert(oCarro["color"]); //despliega "rojo"  
alert(oCarro["puertas"]); //despliega "4"  
alert(oCar["pagado"]); //despliega "true"
```


El mismo objeto también puede ser creado utilizando el constructor de objetos de JavaScript.

```
var oCarro = new Object();
```

```
oCarro.color = "rojo";
```

```
oCarro.puertas = 4;
```

```
oCarro.pagado = true;
```

Es posible también mezclar objetos y vectores, creando un vector de objetos o un objeto que contiene vectores. Por ejemplo:

```
var aCarros = [
```

```
{
```

```
  "color" : "rojo",
```

```
  "puertas" : 2,
```

```
  "pagado" : true
```

```
},
```

```
{
```

```
  "color" : "azul",
```

```
  "puertas" : 4,
```

```
  "pagado" : true
```

```
},
```

```
{
```

```
  "color" : "blanco",
```

```
  "puertas" : 2,
```

```
  "pagado" : false
```

```
  } ];
```

Este objeto es un arreglo con varios objetos dentro de él. Cada uno de los objetos representa un carro distinto. Para acceder a los elementos digamos del carro intermedio utilizamos la notación punto de esta manera:

```
alert(aCarros[1].puertas); //despliega "4"
```

También se puede definir vectores o arreglos dentro de un objeto , de esta manera:

```
var oCarrosInfo = {  
    "coloresDisponibles" : [ "rojo", "blanco", "azul" ],  
    "numeroPuertas" : [ 2, 4 ]  
};
```

Este código define un objeto llamado oCarrosInfo que tiene dos propiedades, cada una de ellas es un arreglo ya que se tiene varios valores.

Para acceder a un valor dentro de esta estructura se procede así:

```
alert(oCarrosInfo.coloresDisponibles[1]);
```

Este código da como resultado el color blanco. Finalmente nos hacemos la pregunta , sabiendo que eso es lo que está pensando exactamente el lector de este proyecto de tesis, Que tiene que ver esto con JSON?. Bueno contestaremos esa pregunta a continuación.

3.5.3 Sintaxis

La sintaxis de JSON no es nada más que la mezcla de notación de objetos y vectores para almacenar datos. La única diferencia radica en que JSON no tiene variables. Recordemos que JSON representa solo datos, no tiene concepto de variables, operadores o igualdades. Por lo tanto, aplicando el último ejemplo, quedaría así:

```
{  
  "coloresDisponibles" : [ "rojo", "blanco", "azul" ],  
  "numeroPuertas" : [ 2, 4 ]  
}
```

Nótese que la variable `oCarrosInfo` ha sido removida, así como también los punto y comas. Si esto se transmitiera vía HTTP a un navegador, sería muy rápido debido al reducido número de caracteres. Supongamos que estos datos fueran recuperados utilizando XMLHTTP o alguna otra forma de comunicación, y almacenado en la variable `JSON`. Se tendría ahora una cadena de información, no un objeto y ciertamente no un objeto con dos vectores o arreglos. Para volver a transformar esto en un objeto, simplemente se utiliza la función `eval()` de JavaScript, de esta manera:

```
var oCarrosInfo=eval("(+ sJSON+)");
```

En este ejemplo se encierra el texto JSON entre paréntesis y luego pasa esa cadena a la función `eval`, que actúa como un intérprete JavaScript. El resultado de esta operación es un objeto JavaScript idéntico al que originalmente se realizó en los ejemplos anteriores. Siendo

este ya un objeto JavaScript genuino , se puede acceder a sus elementos como lo describimos anteriormente.

Ahora podemos ver los beneficios de JSON como formato de datos de JavaScript para Ajax.

Nos ahorra el parsing y podemos rápidamente acceder a los datos contenidos.

3.5.4 Codificación y Decodificación

Como parte de su esfuerzo en la creación de JSON , Douglas Crockford ha creado una utilidad para codificar y decodificar JSON y objetos JavaScript que puede ser encontrada en el siguiente

URL : www.crockford.com/JSON/json.js

Este script nos brinda el método `JSON.parse()` que convierte solo JSON a JavaScript , ya que `eval()` tiene un potencial riesgo de seguridad al no ser exclusivo de JSON sino que puede interpretar cualquier código que se le pase como parámetro. Así que es más recomendable utilizar esta `JSON.parse()` u otra forma segura de codificación.

El método para convertir un objeto JavaScript a JSON es `JSON.stringify()`. Todo lo que se necesita hacer es pasar el objeto a esta función para obtener el JSON necesario para la transmisión de datos.

Consideremos el siguiente ejemplo:

```
var oCarro = new Object();
```

```
oCarro.puertas = 4;
```

```
oCarro.color = "azul";
```

```
oCarro.anio = 1995;
oCarro.drivers = new Array("Penny", "Dan", "Kris");
document.write(JSON.stringify(oCarro));
```

Este código despliega lo siguiente

```
{"puertas":4,"color":"azul","anio":1995,"drivers":["Penny","Dan","Kris"]}
}
```

Este es el JSON listo para ser transmitido al servidor.

3.5.5 JSON vs. XML

Como se mencionó anteriormente, una de las ventajas de JSON sobre XML es que el primero es más compacto y de menor tamaño, lo que se traduce en rendimiento y velocidad óptimos.

Consideremos el siguiente ejemplo:

```
<classinfo>
  <estudiantes>
    <estudiante>
      <nombre>Sixto Vizuite</nombre>
      <promedio>99.5</promedio>
      <edad>17</edad>
      <segradua>>true</segradua>
    </estudiante>
    <estudiante>
      <nombre>Miguel Toaza</nombre>
      <promedio>34.87</promedio>
      <edad>17</edad>
```

```

    <segradua>false</segradua>
  </estudiante>
<estudiante>
  <nombre>Rebeca Saraguro</nombre>
  <promedio>89.6</promedio>
  <edad>18</edad>
  <segradua>true</ segradua >
</estudiante>
</estudiantes>
</classinfo>

```

Este ejemplo contiene información acerca de tres estudiantes de un curso. De una vez podemos reconocer información que no es necesaria en este XML. Los elementos <classinfo> y <estudiantes>. Aunque estos elementos ayudan a definir la estructura y significado de la información, lo que realmente nos interesa es la información de los alumnos, el nombre de la información se repite pero la información que cuenta realmente no se repite. Ahora veamos la misma información en formato JSON.

```

{ "classinfo" :
  {
    "estudiantes" : [
      {
        "nombre" : "Sixto Vizquete",
        "promedio" : 99.5,
        "edad" : 17,
        "segradua" : true
      }
    ]
  }
}

```

```
},  
  
  {  
    "nombre" : "Miguel Toaza",  
    "promedio" : 34.87,  
    "edad" : 17,  
    "segradua" : false  
  },  
  
  {  
    "nombre" : "Rebeca Saraguro",  
    "promedio" : 89.6,  
    "edad" : 18,  
    "segradua" : true  
  }  
]  
  
}  
  
}
```

Como se puede observar en este ejemplo, mucha de la información superflua no está presente. Debido al hecho de que no se requiere etiquetas que cierren otras etiquetas que se abren, grandemente se reduce el número de bytes que se necesita para transmitir la misma información, por eso es que el creador de JSON lo denominó **“Una alternativa libre de grasa al XML”**.

La única desventaja de JSON comparado con XML es que no es tan fácil de leer como su contraparte. Esos extra bytes que tiene XML facilitan la lectura y el entendimiento de la estructura, mientras JSON puede ser difícil de leer. Pero en el mundo real , esto no aplicaría ya

que JSON es generado y decodificado por una herramienta de software, que toma un objeto , lo transforma en JSON y de vuelta al mismo objeto, por lo que no debería ser problema la interpretación , es mas no se la tiene porqué hacer.

3.5.6 Portabilidad

JSON es portable a todas las plataformas de desarrollo y tecnologías existentes. Algunas de estas incluyen:

PHP , .NET , Java , Ruby on Rails , Coldfusion , Perl , Python , etc.

Douglas Crockford mantiene una lista de utilidades JSON en su página web:

www.crockford.com/JSON/index.html.

de todas estas tecnologías y otras más.

3.6 EXTJS

Extjs comenzó como una extensión de la poderosa librería YUI (Yahoo User Interface). Pero Extjs no es otra librería JavaScript más, de hecho es un framework que además puede trabajar conjuntamente con otras librerías JavaScript utilizando adaptadores, en lo que ahondaremos más adelante.

Extjs se debe utilizar en sitios web que requieran un alto nivel de interacción con el usuario, lo cual es algo más complejo que un típico sitio web. Un sitio web que requiera procesos y un flujo de trabajo sería un ejemplo perfecto.

Extjs contribuye al desarrollo de aplicaciones en los siguientes aspectos:

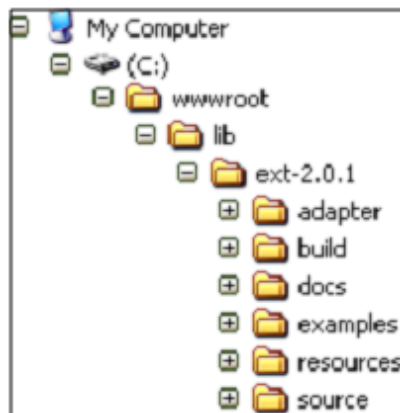
- Provee compatibilidad con todos los browsers que es transparente al desarrollador.
- Hace posible implementar complejos componentes en una página web.
- Ofrece comunicación optimizada con el servidor sin tener que hacer recarga de la página. Es decir implementa Ajax en tiempo real.
- Nos permite elegir entre desarrollo tradicional y el desarrollo utilizando Ajax
- Interfaces manejadas a través de eventos.

3.6.1 Instalación

Para instalar extjs se debe primero bajarse la distribución del sitio de extjs

www.extjs.com

Actualmente existe la versión 3.x que es la más estable. Una vez que se haya bajado la distribución se debe descomprimir en el directorio raíz del servidor Web quedándonos la estructura de directorios como se muestra en la gráfica siguiente:



El SDK contiene una versión de Extjs que tiene todo incluido llamada ext-all. También contiene una versión usada para desarrollo llamada la versión debug, que es la que se utiliza en el ambiente de desarrollo. La versión debug nos hace más fácil localizar errores en el

código debido a que no está comprimida, hecho por el cual nos reportará los errores y las líneas en el código donde estas ocurren. Cuando sea la hora de poner nuestra creación en producción podemos cambiar a que nuestra aplicación utilice ext-all y todo continuará trabajando normalmente.

En el SDK también se encuentra incluida la especificación de dependencias, documentación y ejemplos. Las carpetas adapter y resources son requeridas para que Extjs trabaje correctamente , todo lo demás es para procesos de desarrollo únicamente.

- Adapter: Archivos que permiten que Extjs utilice otras librerías externas.
- Build: Archivos que pueden ser utilizados para construir un ext-all.js customizado.
- Docs: El centro de documentación (solo trabaja en un servidor web)
- Examples: Algunos ejemplos de código fuente
- Resources: Dependencias de la librería Extjs, tales como CSS e imágenes.
- Source: El código fuente de Extjs.

Cuando todo esté listo para subir el sitio a un servidor Web, las carpetas adapter y resources deben ser subidas también.

3.6.2 Inclusión de Extjs

Antes de que podamos utilizar Extjs en nuestras páginas, necesitamos referenciar a las librerías de Extjs para que nuestro proyecto las pueda utilizar. Esto debemos hacerlo en la porción HEAD de nuestra página HTML.

```
<html>
<head>
  <link rel="stylesheet" type="text/css"
    href="lib/extjs/resources/css/ext-all.css" />
  <script src="lib/extjs/adaptor/ext/ext-base.js"></script>
  <script src="lib/extjs/ext-all-debug.js"></script>
</head>
<body>
  <!-- Nothing in the body -->
</body>
</html>
```

El path a los archivos Ext debe ser correcto y además ser relativo a la localización de nuestro archivo HTML. Estos archivos deben incluirse en el siguiente orden:

- ext-all.css: El principal CSS de Extjs , es un archivo de estilos que controla el look n´ feel de los widgets de Extjs. Este archivo debe ser incluido tal como nos viene, no es recomendable realizar modificaciones a este archivo ya que pudiera tener problemas con actualizaciones futuras. Si se tiene que hacer cambios en los estilos, es mejor incluir estos cambios en un archivo .css por separado y realizar los debidos overrides. Este archivo debe ser incluido después de ext-all.css
- ext-base.js: El adaptador de Ext. , provee la funcionalidad del core de Extjs. Este es el archivo que deberíamos cambiar si se quisiera utilizar otra librería como JQuery en conjunto con Ext.

- `ext-all.js` o `ext-all-debug.js`: El archivo principal de la librería Ext, todos los widgets residen en esta archivo. La versión debug es para desarrollo mientras que la otra versión es para el ambiente de producción.
- Un archivo `.css` representando un theme o tema.

Una vez que estos archivos se encuentran en su lugar, podemos empezar a utilizar Extjs.

3.6.3 Uso de Extjs

Vamos a empezar con un ejemplo simple.

```
<html>
<head>
  <title>Ejemplo simple </title>
  <link rel="stylesheet" type="text/css"
    href="lib/extjs/resources/css/ext-all.css" />
  <script src="lib/extjs/adaptor/ext/ext-base.js"></script>
  <script src="lib/extjs/ext-all-debug.js"></script>
  <script>
    <!--El código de Extjs se coloca dentro de una función Ext.onReady -->
    Ext.onReady(function(){
      Ext.Msg.alert('Hello', 'World Example');
    });
  </script>
</head>
<body>
```

```
<!--Nada en el cuerpo de la pagina -->  
  
</body>  
  
</html>
```

El ejemplo anterior despliega lo siguiente:



Figura 3.1 : Aplicación simple Hello Word utilizando Extjs

Como una ventana de dialogo real, se la puede arrastrar por toda el área de la ventana del navegador pero no más allá debido a que no se una ventana de dialogo real, es una colección de etiquetas DIV e imágenes juntadas para imitar una ventana de dialogo. También se puede notar como los botones Ok y Cancelar se iluminan al pasar el cursor por encima. Nada mal para una sola línea de código! Como una nota adicional, es mejor crear un archivo javascript por separado e incluirlo en la pagina HTML.

3.6.4 Adaptadores

Cuando Ext estaba siendo desarrollada, se le dió la opción de utilizar otras dos frameworks a mas de YUI : jQuery o Prototype con Scriptaculous.

Esto significa que si tuviéramos que utilizar otra librería más apropiada para nuestras necesidades, esta se la podría utilizar en conjunto con Ext usando el adaptador apropiado. Ext

continuará trabajando normalmente sin importar que adaptador se incluya. Ext tiene su propio adaptador, que es un adaptador a sí mismo.

Para utilizar un adaptador se debe primero incluir la librería externa correspondiente y luego incluir el archivo adaptador relacionado con esta, el cual se encuentra en la carpeta adapters del SDK. Para utilizar otras librerías solo se debe reemplazar la línea de **include** del adaptador Ext por defecto con las líneas que referencian a las librerías que se va a incluir, como se muestra a continuación

```
<script src="lib/extjs/adapter/ext/ext-base.js"></script>
```

Para jQuery, incluya estas líneas en el area HEAD de la página web:

```
<script src="lib/jquery.js"></script>
```

```
<script src="lib/jquery-plugins.js"></script>
```

```
<script src="lib/extjs/adapter/jquery/ext-jquery-adapter.js">
```

```
</script>
```

Para YUI, incluir estos archivos en el HEAD. El archivo de utilidades está localizado en el directorio build/

```
<script src="lib/utilities.js"></script>
```

```
<script src="lib/extjs/adapter/yui/ext-yui-adapter.js"></script>
```

Para "Prototype + Scriptaculous", se incluye esto en el HEAD:

```
<script src="lib/prototype.js"></script>
```

```
<script src="lib/scriptaculous.js?load=effects"></script>
```

```
<script src="lib/extjs/adapter/prototype/ext-prototype-adapter.js"></script>
```

3.6.5 Internacionalización

Los objetos Extjs pueden ser desplegados en varios lenguajes y actualmente existen más de 40 traducciones. Todas estas traducciones son creadas por la comunidad. Los archivos de internacionalización están incluidos en la distribución. Para utilizar nuestro lenguaje nativo podemos copiar el archivo de lenguaje a la carpeta lib , podemos editarlo y añadir texto o modificar el texto de este archivo como mejor nos convenga para nuestra aplicación. Existen tres escenarios para internacionalización que requieren tres enfoques separados:

- Solo Inglés
- Un solo lenguaje diferente al inglés
- Lenguajes múltiples

Solo Inglés

Esto no requiere modificaciones al setup estándar, y no hay archivos extra a incluir debido a que la traducción en Inglés está ya incluida en el archivo ext-all.js

Un solo lenguaje diferente al Inglés

La segunda opción requiere que incluyamos una de los archivos de idioma del directorio build/locale. Esta opción trabaja sobrescribiendo las cadenas de texto en Inglés por lo que debe incluirse después de todas las librerías.

```
<link rel="stylesheet" type="text/css"
  href="lib/extjs/resources/css/ext-all.css" />
<script src="lib/extjs/adapter/ext/ext-base.js"></script>
<script src="lib/extjs/ext-all-debug.js"></script>
<script src="lib/extjs/build/locale/ext-lang-es.js"></script>
```

En el ejemplo se ha incluido el lenguaje Español. Ahora veamos como se ve nuestro ejemplo aplicado el archivo de Español.

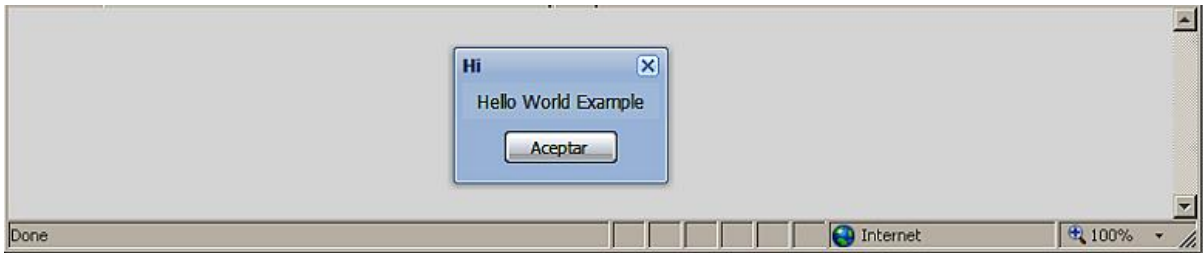


Figura 3.2: Internacionalización de Extjs

Como podemos notar, el botón tiene la etiqueta Aceptar.

Todos los elementos parte de la UI han sido localizados, estos generalmente incluyen: textos de calendario, mensajes de error, mensajes de ayuda contextual e indicadores de carga. Los mensajes que son específicos de la aplicación deben ser traducidos por el desarrollador e incluidos al archivo de lenguaje correspondiente, o añadido a un nuevo archivo. El método preferido es crear un archivo separado que contiene únicamente las adiciones y cambios que se necesita, esto nos deja preparados para arreglos y actualizaciones al archivo principal de lenguaje.

Lenguajes Múltiples

El tercer método de cambiar lenguajes es básicamente el mismo que el segundo. Solo necesitamos añadir algo de código a nuestra página para cambiar entre archivos de lenguaje.

Desafortunadamente, estos cambios no pueden ser realizados dinámicamente. En otras palabras, no podemos hacerlo en tiempo real y ver como ocurre en la pantalla.

3.6.6 La función OnReady

```
Ext.onReady(function(){  
    Ext.BLANK_IMAGE_URL = 'images/s.gif';  
    Ext.Msg.show({  
        title: 'Estimado Usuario',  
        msg: 'Texto del Mensaje',  
        buttons: {  
            yes: true,  
            no: true,  
            cancel: true  
        }  
    });  
});
```

La función `onReady` es lo que utilizamos para hacer que nuestro código se ejecute previa espera a que el documento principal esté cargado. El argumento pasado a `onReady` es una función, la cual puede ser un nombre de una función o una función in-line.

3.6.7 JSON y los objetos de configuración

En Extjs se utiliza objetos de configuración para los diferentes componentes que utilizan la notación JSON.

Usando objetos de configuración somos capaces de tener un mayor nivel de flexibilidad. El orden de los argumentos no importa. Con el objeto de configuración al pasar los parámetros a funciones, los argumentos ya no necesitan estar atados a un lugar específico.

```
var test = new TestFunction({  
    firstWord: 'three',  
    secondWord: 'fixed',  
    thirdWord: 'arguments'  
});
```

Este método permite una ilimitada expansión de los argumentos de nuestras funciones. Usar menos argumentos o añadir nuevos argumentos es simple.

Otra gran ventaja de utilizar objetos de configuración es que las llamadas anteriores a las funciones no serán afectadas por la adición o substracción de argumentos a un punto más tarde.

```
var test = new TestFunction({  
    secondWord: 'three'  
});
```

```
var test = new TestFunction({
```

```
secondWord: 'three',  
fourthWord: 'wow'  
});
```

3.6.8 Objetos de configuración

Si se está familiarizado con CSS o JSON nos podemos dar cuenta que el objeto de configuración se ve similar a ambos CSS y JSON, y en verdad son lo mismo. Los objetos de configuración son solo maneras de estructurar los datos para que puedan fácilmente ser leídos por los diferentes lenguajes de programación. En nuestro caso JavaScript.

Echemos un vistazo a este ejemplo:

```
{  
  title: 'Milton',  
  msg: 'Deseas ingresar al siguiente nivel?',  
  buttons: {  
    yes: true,  
    no: true,  
    cancel: true  
  },  
  icon: 'milton-icon',  
  fn: function(btn) {  
    Ext.Msg.alert('Usted eligió , btn);  
  }  
}
```

Esto genera un panel con tres botones asociados a una función para manejar la elección realizada por el usuario. Los componentes se manejan con objetos de configuración, y es por eso que es importante tomarse un tiempo para aprenderlos.

A continuación se lista algunos consejos claves para utilizar objetos de configuración:

- Las llaves encierran todo el conjunto de registros.
- Cada registro consiste de un set de pares en la forma nombre-valor separados por dos puntos (:) y los pares separados por comas, como por ejemplo: —{nombre0: valor0, nombre1: valor1 }
- Los valores de los registros pueden contener cualquier tipo de datos incluyendo boolean , vectores, funciones o hasta otro objeto. Ejemplo:
 - {nombre0: true, nombre1: { nombre2: valor2 } }
- Los corchetes [] identifican un vector o arreglo.
Ejemplo: {name: [uno, dos, tres] }.

Una ventaja de usar JSON para configurar objetos es que si queremos mas opciones, solo tenemos que escribirlas y listo. A diferencia de la típica llamada a una función, el orden de los parámetros de configuración es irrelevante y pueden haber tantos como se necesite.

Más información sobre los componentes y otros muchos asuntos concernientes pueden ser encontrados en el sitio de Extjs. La extensa documentación es el mejor sitio de aprendizaje y está disponible en la siguiente dirección:

www.extjs.com/deploy/ext/docs/

Además existen los fórums donde miembros de la comunidad intercambian información en la siguiente dirección:

www.extjs.com/forum/

CAPITULO IV

DESARROLLO DEL CASO DE ESTUDIO

4.1 CARACTERÍSTICAS DEL SISTEMA

A continuación se va a desarrollar un sistema con las siguientes características:

- Metodología: **RUP**
- BackEnd: **Mysql 5.0.45**
- Framework: **Extjs 3.1**
- Lenguaje del Servidor: **PHP 5 Orientado a objetos**
- Servidor: **Apache 2.x**
- Persistencia de BD: **Active Record**
- Formato de Transferencia de datos entre cliente y servidor: **JSON**
- Arquitectura : **MVC : Model View Controller**

4.2 METODOLOGIA DE DESARROLLO

El proyecto se desarrolló con la metodología de desarrollo denominada RUP, la cual brinda un marco de referencia de miles de proyectos de todo el mundo con las mejores prácticas. A pesar que esta metodología tiene una cantidad de artefactos, sin embargo, es una buena práctica el utilizar únicamente los que brinden valor al proyecto. Es por eso que RUP se puede adaptar a proyectos pequeños y medianos. En esta sección del capítulo se presenta los artefactos o entregables generados para el presente proyecto dividido en las fases que son Inicio, Elaboración, Construcción y Transición.

4.3 FASE DE INICIO

Durante la fase de inicio se tiene una carga de trabajo más pronunciada en lo que es la disciplina de requerimientos.

FASE: INICIO	ID: INI001
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCION:	Plan de Desarrollo de Software
VERSION:	1.0

© 2010 Lincoln College of Foreign Languages
 TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.
 No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

4.3.1 Plan de desarrollo del software

1. Introducción

En este documento se establecen los aspectos claves para la administración y gestión adecuada del desarrollo del proyecto presente. Con esto se pretende brindar un marco de trabajo, con el cual delimitar las responsabilidades y compromisos para el éxito del proyecto. El detalle de las actividades para llevar a cabo los objetivos del proyecto se muestran en el cronograma.

1.1 Propósito

El propósito de este Plan de Desarrollo de Software es definir el desarrollo de actividades en términos de las fases e iteraciones requeridas, destinado al proyectar una nueva imagen del Lincoln College of Foreign Languages si este proyecto fuera implementado.

1.2 Alcance

Este Plan de Desarrollo de Software describe el plan total para ser usado por el Sistema Escolástico Lincoln College. Los detalles de las iteraciones individuales serán descritos en los planes de iteración. Los planes descritos en este documento son basados en las exigencias de producto definidas en el documento de Visión.

1.3 Definiciones, Acrónimos, y Abreviaturas

- RUP: Rational Unified Process.
- Workflow: Flujo de trabajo
- NA: No Aplica

1.4 Descripción

Este Plan de Desarrollo de Software contiene la información siguiente:

- **La descripción del proyecto** - proporciona una descripción del propósito del proyecto, el alcance, y objetivos. Esto también define los entregables del proyecto.
- **La organización de proyecto** - describe la estructura de organización del equipo del proyecto.
- **El proceso de Gestión** - explica el costo estimado y el Cronograma, define las fases principales y metas para el proyecto, y describe como el proyecto será supervisado.

1.5 Propósito del Proyecto, Alcance, y Objetivos

El propósito del proyecto es construir un Sistema que permita la automatización de los procesos de un sistema escolástico para que sus clientes internos y externos estén informados y puedan realizar sus trámites más convenientemente.

1.6 Etapas del Proyecto

Las etapas del proyecto se enmarcan en el contexto del RUP - Rational Unified Process o Proceso Unificado de Desarrollo de Software con el cual se busca:

- Desarrollar de forma iterativa
- Administrar los riesgos (en las iteraciones tempranas del desarrollo)
- Dirigir el desarrollo por casos de uso y administrar requerimientos, con los cuales se gerencia el alcance, las expectativas y se cumpla con las necesidades de los afectados – *stakeholders*
- Desarrollar basado en arquitectura, con soporte de modelación con UML – Unified Modeling Language.

La siguiente tabla muestra las etapas del RUP y una descripción genérica del enfoque de cada una:

Tabla 4.1: Descripción de las fases del RUP

Etapa	Objetivo
Conceptualización	Obtener un acuerdo de todos los afectados claves en el alcance y factibilidad con base en el conocimiento del modelo de casos de uso del sistema y los riesgos del proyecto. Se toman decisiones en las estrategias de mitigación de los riesgos. Se modela el negocio para determinar como el sistema debe soportar el negocio.
Elaboración	Establecer la arquitectura base del sistema la cual proveerá un fundamento estable y la definición de las principales decisiones de diseño, que guiarán los esfuerzos de diseño y desarrollo de los casos de uso y el cumplimiento de las especificaciones suplementarias durante la fase Construcción. Esta arquitectura se definirá tomando en cuenta los requerimientos más significativos (los que tienen impacto en la arquitectura del sistema) y una valoración del riesgo.
Construcción	<p>Desarrollar el sistema basado en la arquitectura base establecida en la fase de elaboración y cumplir con los requerimientos establecidos y delimitados tanto por los usuarios como por el equipo de trabajo del proyecto.</p> <p>En esta etapa se toman en cuenta las consideraciones necesarias para administrar el alcance para poder realizar salidas parciales con la aplicación y poder tener un producto (parcial pero funcional) para la restricción de tiempo solicitada por el negocio.</p>
Transición	Liberación del Sistema a la comunidad de usuarios, cumpliendo con los requerimientos plasmados en el modelo de casos de uso y las especificaciones suplementarias de software.

1.7 Entregables por Fase

En esta sección se listan los entregables propuestos en cada una de las fases del proceso unificado de desarrollo de software (Rational Unified Process). Lo cual se encuentra en forma específica y detallada, en el contrato celebrado entre las partes

Tabla 4.2: Entregables por fase

Fase	Entregables
Conceptualización	<ul style="list-style-type: none">• Visión de Negocio• Modelo de Casos de Uso de Negocio• Especificación de Casos de Uso de Negocio• Modelo de Casos de Uso de Sistema• Especificaciones Suplementarias• Lista de Riesgos• Plan de Desarrollo de Software
Elaboración	<ul style="list-style-type: none">• Plan de Elaboración• Documento de Arquitectura SAD (Software Architecture Document).
Construcción	<ul style="list-style-type: none">• Código fuente• Plan de Pruebas• Resultado de las Pruebas• Manual de Instalación
Transición	<ul style="list-style-type: none">• Código Fuente• Capacitación

1.8 Iteraciones

A continuación se describen las iteraciones propuestas y de acuerdo a lo establecido de forma inicial.

Tabla 4.3: Iteraciones por fase

Etapa	Iteración	Objetivo	Alcance (requerimientos)
Conceptualización	Iteración I1	Establecer el alcance del sistema mediante un modelo de casos de uso, la identificación de los riesgos y la modelación de los procesos de negocio que permita determinar las mejoras en los procesos actuales y que se soportarán por el sistema	<ul style="list-style-type: none"> Casos de uso del negocio Visión del negocio Lista de riesgos Especificación del casos de uso del negocio
Elaboración	Iteración E1	Establecer las principales decisiones de diseño del sistema basado en el análisis de arquitectura de los casos de uso y especificaciones suplementarias que son críticas que tienen riesgos o complejidad técnica y que guiarán la fase de construcción.	<ul style="list-style-type: none"> • Plan de elaboración
	Iteración E2	Afinar la arquitectura propuesta y solucionar mecanismos o riesgos técnicos restantes.	<ul style="list-style-type: none"> • Documento de Arquitectura SAD (Software Architecture Document) •
Construcción	Iteración C1	Construcción de acuerdo a especificación de casos de uso.	<ul style="list-style-type: none"> • Ingresar y Mantener Datos • Mantener Parámetros Corporativos • Generar Reportes y

			Estadísticas
	Iteración C2	Construcción de acuerdo a especificación de casos de uso.	<ul style="list-style-type: none"> • Matricular Alumnos • Crear Curso • Asignar Profesor
	Iteración C3	Construcción de acuerdo a especificación de casos de uso.	<ul style="list-style-type: none"> • Gestionar Calificaciones • Ingresar Asistencia
	Iteración C4	Construcción de acuerdo a especificación de casos de uso. Realizar pruebas y documentar mediante casos de pruebas derivados de los casos de uso del sistema.	<ul style="list-style-type: none"> • Generar Reporte de Asistencia • Generar Rol de Pagos • Casos de prueba • Creación de Manual de Instalación
Transición	Iteración T1		<ul style="list-style-type: none"> • Creación de Manual de Usuario
	Iteración T2		<ul style="list-style-type: none"> • Código fuente listo

1.9 Supuestos y Restricciones

El proyecto es parte de un proyecto de tesis aprobado por la Carrera de Ingeniería de Sistemas e Informática del Departamento de Ciencias de la Computación de la Escuela Politécnica del Ejército como caso de estudio. Debido a lo anterior y las limitaciones de hardware e infraestructura dentro de la empresa auspiciante no se pone el sistema en producción.

1.10 Estimaciones del Proyecto

A continuación se resumen las fechas de inicio y finalización de las macro-actividades del proyecto. El detalle de las actividades puede ser consultado en el cronograma (en *MS - Project*).

Tabla 4.4: Estimación de tiempos del proyecto.

Etapa	Duración en Días Hábles (Naturales)	Duración en Semanas	Fecha de Inicio	Fecha de Finalización
1. Conceptualización	10 (15)	2	Miércoles 16 Noviembre 2009	Martes 29 Noviembre 2009
2. Elaboración	27 (33)	4	Miércoles 30 Noviembre 2009	Viernes 13 Enero 2010
7. Construcción	42 (73)	10	Lunes 16 Enero 2010	Viernes 17 de Marzo 2010
8. Transición	12 (15)	2	Lunes 20 de Marzo 2010	Martes 4 de Abril 2010
Totales	91 (136)	18		

Ver (Anexo D) : Diagrama de Gantt del Proyecto

Tabla 4.5: Total de Meses del Proyecto

Etapa	Duración en Meses
1. Conceptualización	0.5
2. Elaboración	1.25
7. Construcción	2
8. Transición	0.5
Total de Meses	4.5

FASE: INICIO	ID: INI002
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCION:	Documento Visión del Negocio
VERSION:	1.0

© 2010 Dirección de Desarrollo Institucional del IESS
 TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.
 No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

4.3.2 Visión de Negocio

1. Introducción

1.1 Propósito

Este documento tiene como propósito establecer el alcance de la problemática de negocio y las necesidades que se deben suplir, así como identificar cuáles de esas necesidades pueden ser suplidas mediante el desarrollo de software y cuáles se solucionarán por otros mecanismos.

1.2 Alcance

El documento describe la visión del negocio del Instituto de Idiomas Lincoln College of Foreign Languages en miras a mejorar su funcionamiento, imagen y servicio a los clientes.

1.3 Definiciones, Acrónimos y Abreviaturas

Afectado (stakeholder): es aquella persona que tiene la visión del sistema o parte de él y que pueden tomar decisiones con respecto al enfoque o rumbo de éste. Por lo que es necesario tomar en cuenta su opinión para la definición de los requerimientos del sistema.

Usuario: se entiende por usuario aquellas personas que utilizan de forma directa el sistema.

1.4 Referencias

Planning a Project with the IBM Rational Unified Process, David West, IBM , 2004

Best Practices for Software Development Teams, Rational Software White Paper, 2001

The Ten Essentials of RUP, Leslee Probasco, Rational Software White Paper, 2001

2. Posicionamiento

2.1 Oportunidades del Negocio

- Minimizar al máximo el riesgo de pérdida de datos.

- Brindar un servicio de excelencia a los usuarios internos y externos.

- Ayudar a la toma de decisiones gerenciales.

- Agilizar los procesos de matriculación y gestión de cursos.

2.2 Definición del Problema

El problema de	La ausencia de una solución informática que permita recoger y mantener datos importantes de los procesos críticos de la Institución, lo cual ha ocasionado pérdida de información y tiempo.
Afecta a	<ul style="list-style-type: none">• Clientes presentes y futuros.• Empleados que deben realizar el doble del trabajo.• La Institución
Cuyo impacto es	Perdida de datos por procesos manuales propensos a errores. Molestias a los clientes. Prestigio disminuido. Desorden.
Una solución exitosa es	<ul style="list-style-type: none">• Confiable• Disponible cuando se la necesite• Eficiente• Eficaz• Fácil de mantener• Libre de formalidades innecesarias.• Segura• Amigable con el usuario• Con tiempos de respuesta óptimos

2.3 Lo que no cubre la solución

Proceso / Actividad no cubierta	Descripción Breve
Facturación	Existen sistemas en el mercado para esto, solo se trabajaría en la Integración de ambos sistemas.
Instalación de Hardware y Redes	La instalación de hardware y redes de comunicaciones está fuera del alcance del proyecto ya que no se va implementar en producción.
Cableado	La instalación de cableado no es parte de este proyecto

3. Descripciones de Afectados y Usuarios

3.1 Resumen de los Afectados

Tabla 4.6: Afectados

Nombre	Descripción	Responsabilidades
Cliente	Es la persona que acuden al instituto para aprender uno o más idiomas	Asistir a clase Tener un rendimiento aceptable
Empleado	Es la persona que interactúa con el cliente y realiza las tareas operativas diarias del sistema.	Tareas operativas del negocio de la institución
Profesor	Son las personas que imparten la enseñanza de diferentes idiomas.	Labores del docente

Gerente	Persona que cumple tareas de supervisión y administración que toma decisiones importantes para el buen funcionamiento de la Institución y su progreso diario.	Toma de decisiones Supervisión
----------------	--	-----------------------------------

3.2 Resumen de usuarios o trabajadores de negocio

La siguiente es la lista de los roles que utilizarán el sistema

Tabla 4.7: Usuarios o Trabajadores del Negocio

Nombre	Descripción
Administrador	Realiza el mantenimiento de parámetros corporativos Genera reportes estadísticos Crea Modifica Roles y Usuarios
Operador	Realiza el ingreso y mantenimiento de datos de clientes y profesores Crea cursos Asigna profesores y horarios a cursos Ingresa Alumnos a curso Genera reportes de asistencia
Profesor	Ingresa asistencia Mantiene calificaciones Cierra cursos

3.3 Principales necesidades de los Afectados / Usuarios

Tabla 4.8: Necesidades de Afectados y Usuarios

Necesidad		Beneficio
1. <i>Apoyo a los procesos del negocio con una plataforma informática que permita el almacenamiento y acceso seguro a los datos.</i>	C	La disponibilidad de una plataforma robusta de alto rendimiento y confiabilidad para que los procesos críticos del negocio fluyan.
2. <i>Proceso de matriculación</i>	C	Proceso de matriculación controlado , rápido y fácil de ejecutar.
3. <i>Gestión de cursos</i>	C	Manejar asuntos relacionados con la gestión de cursos.
4. <i>Gestión de calificaciones</i>	I	Manejar asuntos relacionados con la gestión de calificaciones.
5. <i>Generar reportes</i>	I	Generación de reportes para toma de decisiones.
6. <i>Rol de pagos profesores</i>	I	Calculo de salarios de profesores.

Ver Notación de Necesidades en el Sección 1: Atributos de las Necesidades

3.4 Metas a Alcanzar con el Modelado de Negocio

Tabla 4.9: Metas a Alcanzar con el Modelado del Negocio

Tipo de Meta	Actual	Esperado
Reducir tiempos de inscripción y matriculación	1 hora	10 minutos
Aumentar el número de cursos abiertos	Promedio de 8	Más de 20 cursos
Optimizar el tiempo en realización de reportes de asistencia y calificaciones	Asistencia 1/2 hora cada curso Calificaciones 1 hora cada curso	Asistencia 2 minutos cada curso Calificaciones 2 minutos cada curso

4. Análisis de Impacto

La aplicación del presente proyecto permitiría una mejora notable para el instituto de lenguajes Lincoln College of Foreign languages promoviendo el orden y la productividad. Para eso deben desecharse ciertas prácticas como por ejemplo: el registro de grandes cantidades de papeleo que se tenía con los procesos manuales y así adaptarse al nuevo esquema.

5. Otros Requerimientos

5.1 Base legal

Reglamento General de los Institutos Superiores y técnicos del Ecuador emitido por el Conesup. (ANEXO A)

Sección 1: Atributos de las Necesidades

Beneficio

Métrica utilizada para clasificar las necesidades. Indica el beneficio que brindará dicho elemento desde la perspectiva del negocio. La escala utilizada se describe a continuación:

- C - Crítico: Elementos o necesidades esenciales. Si no se cumplen con ellos, el negocio no podrá realizarse, ocasionando problemas graves.
- I - Importante: Son elementos importantes para la eficacia del negocio, el negocio no puede operar fácilmente si no se suple esta necesidad, sin embargo el negocio podrá seguir funcionando. La carencia de un elemento importante puede afectar la satisfacción del cliente, del usuario, sin embargo la liberación de una solución no será retrasada debido a la carencia de alguna necesidad importante.
- U - Útil: Necesidades que son de valor agregado. Su presencia brindará facilidades de valor agregado al negocio. Sin embargo, el no contar con estos elementos no afectará la satisfacción del usuario/cliente

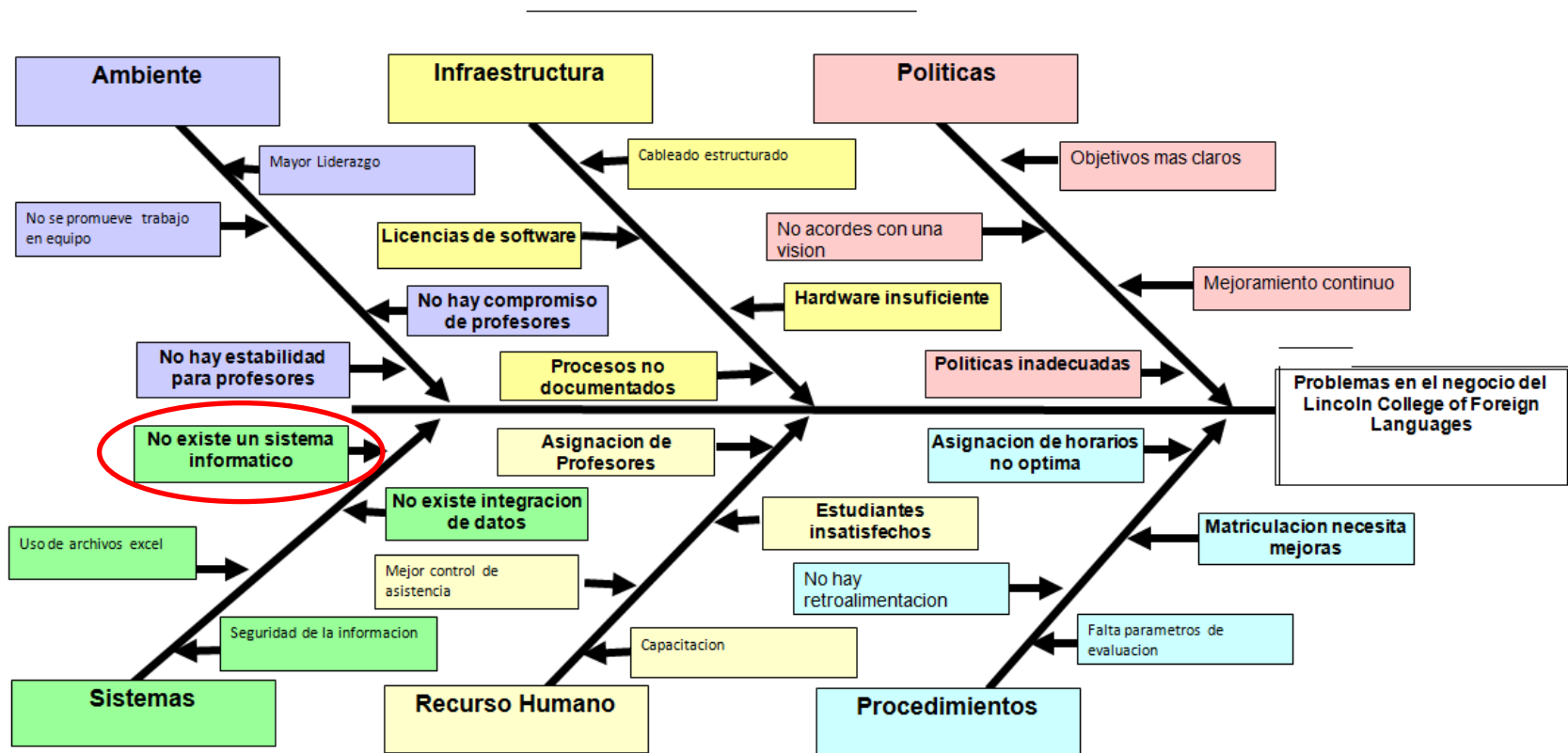
Sección 2: Lluvia de Ideas de la Problemática (Técnica de “Espina de Pescado”)

Tabla 4.10: Problemas / Puntos de Dolor - Lluvia de Ideas

Problemas	Causas
1. La búsqueda de datos de un estudiante toma mucho tiempo	1. Como para cada alumno se abre un archivo Excel, al tener muchos archivos de este tipo, estos se confunden lo cual dificulta la búsqueda.
2. El proceso de inscripción es lento y desordenado	2. Dentro de las hojas de Excel creadas para cada alumno se guarda los datos de inscripción de una manera desordenada
3. El proceso de matriculación no es eficiente	3. El proceso de matriculación se hace en varios pasos. Primero el estudiante se inscribe, luego se pone en espera hasta que hayan 3 o más alumnos para el mismo nivel (en el caso de modalidad grupal), o hasta que se adapte manualmente un horario de un profesor disponible o se haga una contratación por horas, lo cual ha ocasionado que algunos de los estudiantes prospectivos desistan la matricula.
4. La asignación de profesores no es optima e influye negativamente en el proceso de matriculación y gestión de cursos	4. No existe constancia de las horas disponibles de los profesores. Para crear cursos se debe llamar y pedir a los profesores que tomen ese horario, lo cual a veces no es posible y se pone en espera a los estudiantes hasta conseguir uno.

<p>5. El mantenimiento de las calificaciones es ineficiente y propenso a cambios no autorizados y perdida de datos</p>	<p>5. Cada profesor mantiene una hoja Excel con los aportes y sus calificaciones. Al final del periodo deben entregar un medio magnético con la información la cual es archivada en una computadora bajo una estructura de directorios.</p>
<p>6. La generación de reportes toma mucho tiempo y esfuerzo</p>	<p>6. Para generar cualquier reporte o estadística se debe recurrir a los archivos Excel individualmente.</p>
<p>7. La seguridad de datos es inexistente</p>	<p>7. Solo se confía en la clave del sistema operativo de la computadora donde están almacenados los datos.</p>
<p>8. La asistencia de estudiantes es difícil de mantener</p>	<p>8. La asistencia se guarda en hojas Excel y se entrega a la Secretaria donde se organiza para la impresión. Esto se añade a las demás tareas que están dispersas.</p>

Sección 2.1 : Análisis del Problema (Diagrama de Ishikawa)



4.3.3 ESPECIFICACION DE CASOS DE USO DEL NEGOCIO

Se identificaron cuatro casos de uso del negocio y son los siguientes:

1. Gestionar Información
2. Gestionar Matriculas
3. Gestionar Calificaciones
4. Gestionar Asistencia

FASE: INICIO	ID: INI003
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCION:	Especificación del Caso de Uso del Negocio: Gestionar Información
VERSION:	1.0

© 2010 Lincoln College of Foreign Languages
 TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Especificación de Caso de Uso del Negocio:

Gestionar Información

Nombre del Caso de Uso del Negocio

Breve Descripción

Este caso de uso describe las actividades que se lleva a cabo dentro del ámbito de gestión de la información.

Metas

Tipo de Meta
Asegurar la integridad , seguridad, confiabilidad de los datos generados por las operaciones diarias
Disponer de la información como medio de apoyo a la toma de decisiones así como también que satisfaga las necesidades de información de los clientes internos y externos.

Objetivos de Desempeño

Objetivo de desempeño 1

Asegurar la integridad , seguridad, confiabilidad de los datos generados por las operaciones diarias	10%	100%
--	-----	------

Objetivo de desempeño 2

Disponer de la información como medio de apoyo a la toma de decisiones así como también que satisfaga las necesidades de información de los clientes internos y externos.	20%	100%
---	-----	------

Flujo de Eventos

A continuación se realiza breve descripción del caso de uso, en la sección 11.2 se encuentra la descripción detallada mediante un diagrama de actividades.

Flujo Básico

El empleado que actúa como representante de la empresa solicita a la persona su información personal que incluye datos personales, datos de contactos, información de disponibilidad en el caso de profesores. Ver FA1

El empleado ingresa los datos personales de la persona y contactos. Ver FA2, FA3, FA4

Se valida que los datos sean los correctos.

Finaliza el caso de uso

Flujos Alternos

FA1. <El empleado consulta los datos de una persona >

El empleado busca a la persona por cedula o por nombre.

El empleado accede a dicha información y la utiliza según sea requerido

FA2. <El empleado genera reportes >

En caso de no ingresar o modificar personas, el empleado puede generar reportes de asistencia de alumnos y cuadros de calificaciones. El gerente como administrador puede generar reportes: roles de pago, estadísticos, asistencia.

FA3. <El empleado actualiza los datos>

El empleado actualiza los datos y guarda esos cambios.

FA4. <El empleado ingresa los datos de un profesor>

El empleado ingresa la disponibilidad de horarios del profesor

El empleado ingresa los idiomas que el profesor impartirá

El empleado ingresa el tipo de posición y el salario del profesor

Reglas de Negocio

RN1. <Ingreso y mantenimiento de datos>

Se almacena información acerca de personas. Estos datos son ingresados al momento de la inscripción, aunque pudieron haber estado allí antes. Así también se puede ingresar profesores

los cuales deben brindar datos de disponibilidad de horarios, idiomas que enseña y la aspiración salarial.

Riesgos

El mayor riesgo es que se cambie los datos sin autorización o que estos queden desactualizados por cambio de situación de los alumnos.

Dueño del Proceso

El dueño del proceso de negocio es el empleado.

Requerimientos Especiales

Referirse al documento “Especificaciones suplementarias de software”

Precondiciones

< Precondición Uno >

El empleado debe estar autorizado para cumplir con estas tareas.

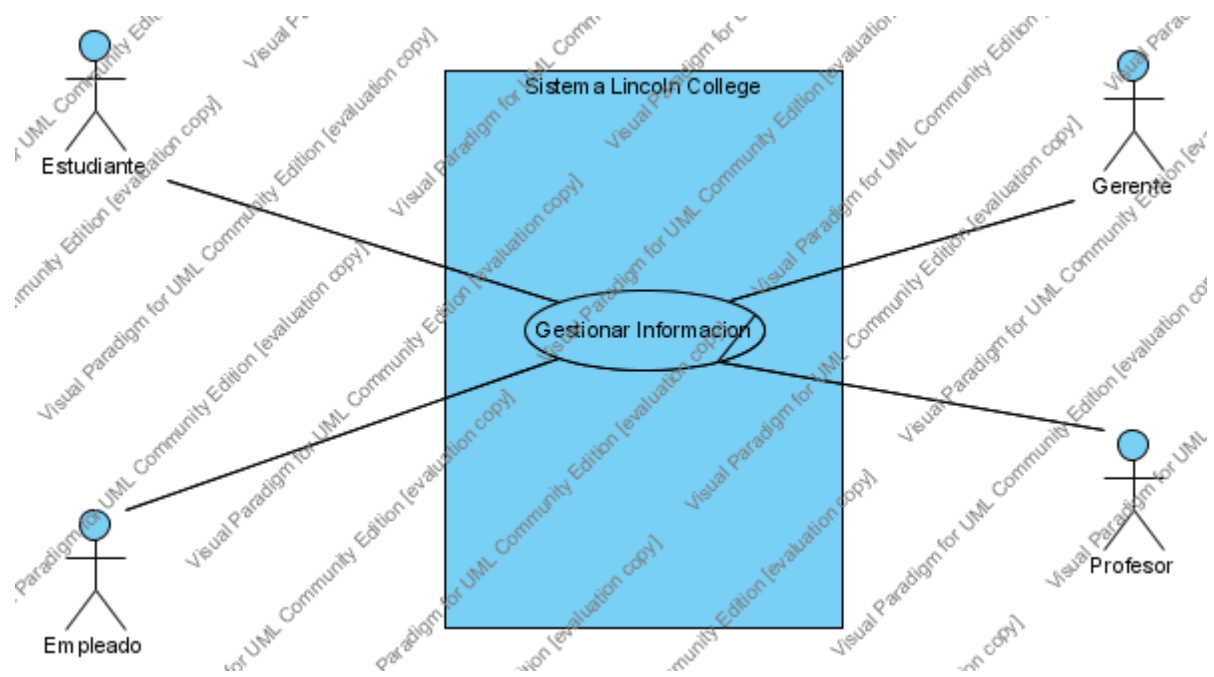
Poscondiciones

< pos condición Uno >

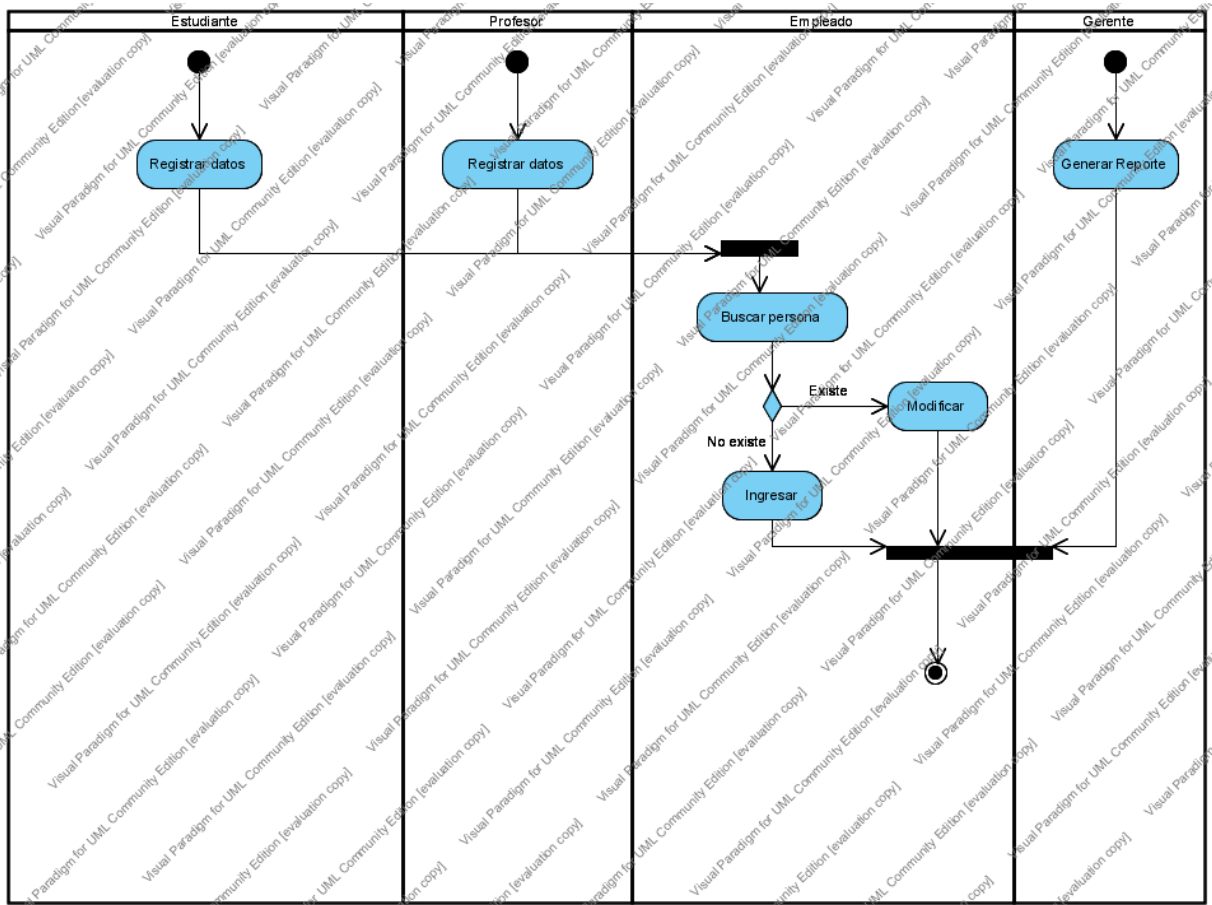
Los datos se encuentran registrados correctamente.

SECCIONES

Sección 11.1 Diagrama de Casos de Uso del Negocio



Sección 11.2 Diagrama de Actividades del Negocio



FASE: INICIO	ID: INI004
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCION:	Especificación del Caso de Uso del Negocio: Gestionar Matrículas
VERSION:	1.0

©2010 Lincoln College of Foreign Languages

TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Breve Descripción

Este caso de uso describe las actividades que se ejecuta para la gestión de matriculas.

Metas

Tipo de Meta
Acelerar el proceso de matriculación, eliminando demoras innecesarias e incómodas para los estudiantes.
Reducir el número de estudiantes que desisten de matricularse por no haber disponibilidad de profesores a los horarios que ellos necesitan.

Objetivos de Desempeño

Objetivo de desempeño 1

Acelerar el proceso de matriculación , eliminando demoras innecesarias e incómodas para los estudiantes	20%	80%
---	-----	-----

Objetivo de desempeño 2

Reducir el número de estudiantes que desisten de matricularse por no haber disponibilidad de profesores a los horarios que ellos necesitan.	20%	80%
---	-----	-----

Flujo de Eventos

A continuación se realiza breve descripción del caso de uso, en la sección 11.2 se encuentra la descripción detallada mediante un diagrama de actividades.

Flujo Básico

El alumno acude al instituto para realizar una matriculación en un curso del idioma de su predilección.

El empleado busca los datos de la persona. Ver FA1

El empleado contacta al profesor para confirmación de disponibilidad. Ver FA2

El empleado crea un curso y asigna un profesor disponible a este.

El empleado ingresa alumnos al curso recién creado.

Finaliza el caso de uso.

Flujos Alternos

FA1. <El empleado no puede encontrar los datos de la persona>

El empleado ingresa los datos de la persona antes de matricularse.

FA2. <El profesor desiste la asignación >

El empleado procede a contactar a otro profesor que tiene ese horario disponible

Si no existen profesores se pide la autorización al alumno para colocarlo en lista de espera hasta que se libere o contrate un recurso.

Reglas de Negocio

RN1. <Proceso de Matriculación>

Se puede matricular uno o varios alumnos en un solo curso, pero un alumno puede estudiar uno o más idiomas al mismo tiempo. Así mismo un curso puede tener uno o más profesores.

El alumno previo a su ingreso en un curso creado debe haber cancelado los valores de no el empleado no lo ingresa.

Primero el empleado crea un curso, luego asigna un profesor disponible y por ultimo ingresa uno o más alumnos al curso creado.

Riesgos

Un riesgo importante es el no disponer de profesores para los horarios que los alumnos solicitan lo cual crea tiempos de espera y la pérdida de negocio si el problema no se soluciona rápidamente. Los profesores son contratados por horas por lo que se debe realizar contrataciones de planta para mitigar el riesgo.

Dueño del Proceso

El dueño del proceso de negocio es el empleado.

Requerimientos Especiales

Referirse al documento “Especificaciones suplementarias de software”

Precondiciones

< Precondición Uno >

El empleado debe estar autorizado para cumplir con estas tareas.

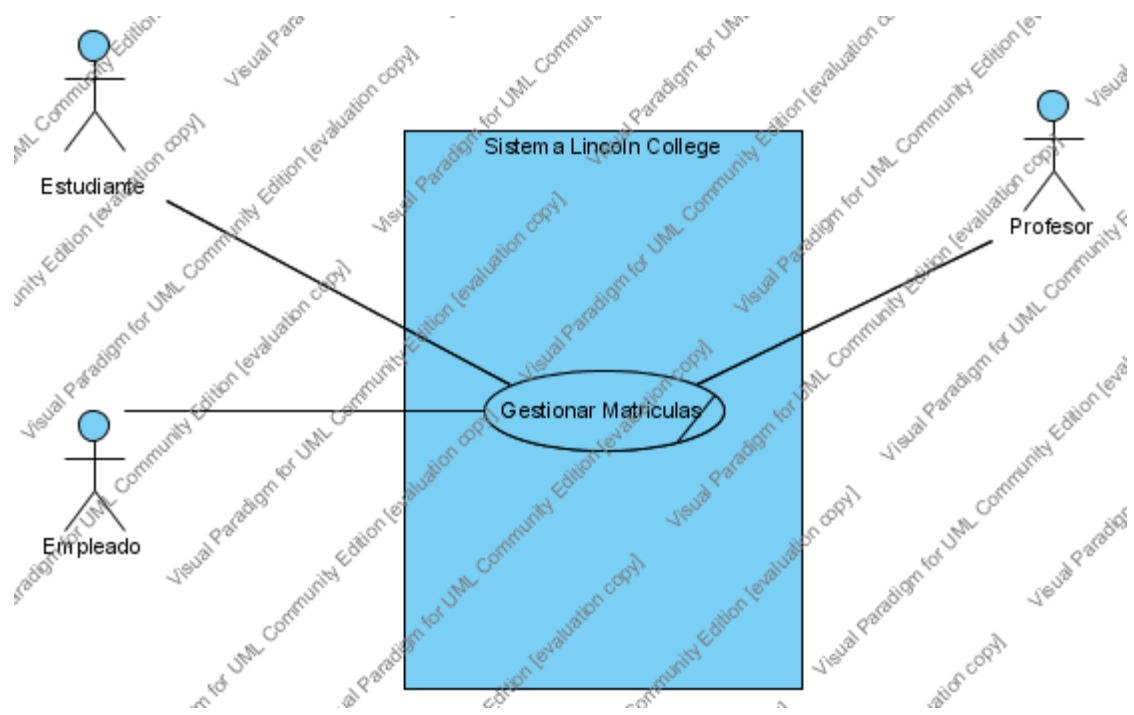
Poscondiciones

< pos condición Uno >

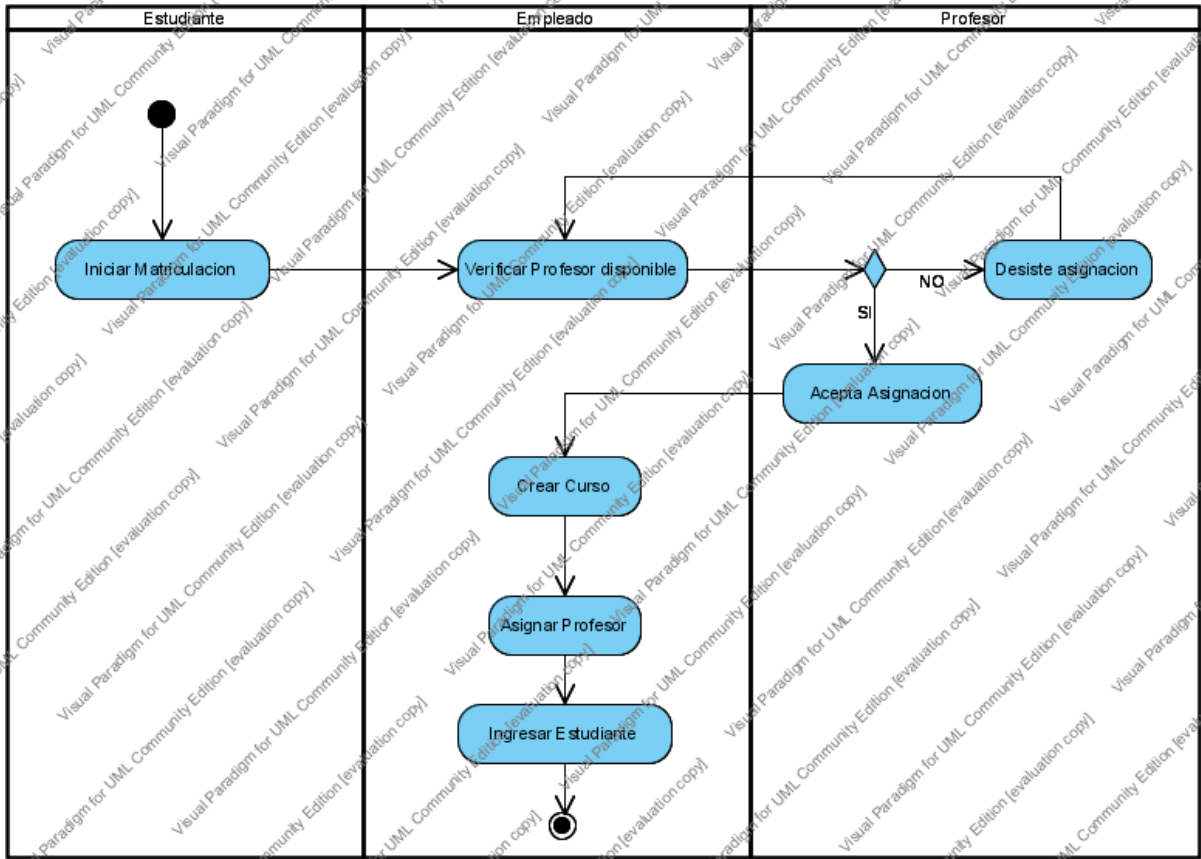
Los datos se encuentran ingresados correctamente.

SECCIONES

Sección 11.1 Diagrama de Casos de Uso del Negocio



Sección 11.2 Diagrama de Actividades del Negocio



FASE: INICIO	ID: INI005
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCION:	Especificación del Caso de Uso del Negocio: Gestionar Calificaciones
VERSION:	1.0

©2010 Lincoln College of Foreign Languages

TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Especificación de Caso de Uso del Negocio: Gestionar Calificaciones

Nombre del Caso de Uso del Negocio

Breve Descripción

Este caso de uso describe las actividades que se ejecuta para la gestión de

Metas

Tipo de Meta
Optimizar el proceso de ingreso y mantenimiento de calificaciones

Objetivos de Desempeño

Objetivo de desempeño 1

Mejorar el proceso de registro y mantenimiento de calificaciones.	20%	90%
---	-----	-----

Flujo de Eventos

A continuación se realiza breve descripción del caso de uso, en la sección 11.2 se encuentra la descripción detallada mediante un diagrama de actividades.

Flujo Básico

El profesor selecciona un curso

El profesor crea un aporte Ver FA1, FA3

El profesor ingresa el valor de la calificación para el aporte Ver FA2, FA3

El profesor entrega el cuadro a Secretaría.

Finaliza el caso de uso

Flujos Alternos

FA1. <El aporte ya existe>

Si el aporte ya fue creado anteriormente el profesor solo puede modificarlo o crear un aporte nuevo.

FA2. <EL profesor modifica un valor ya existente >

El profesor ingresa el nuevo valor de la calificación sobrescribiendo el anterior

FA3. <El curso ya está cerrado >

Si el curso está cerrado , no se puede modificar calificaciones

Reglas de Negocio

RN1. <Proceso de Matriculación>

Según las políticas de la institución se han creado varios aportes a partir de los cuales los estudiantes serán evaluados mensualmente. Estos aportes son evaluados sobre 20 puntos , de los cuales una calificación de 16 se considera como suficiente para aprobar .

El profesor puede modificar dichas calificaciones hasta cerrar el curso en donde ya quedan registradas permanentemente.

Riesgos

Un riesgo importante es el que se pueda modificar las notas por descuido del profesor al dejar su máquina sin bloqueo o no cuide su clave de acceso. Por eso para mitigar este riesgo se enfatiza a los profesores que no presten a nadie su clave.

Dueño del Proceso

El dueño del proceso de negocio es el profesor.

Requerimientos Especiales

Referirse al documento “Especificaciones suplementarias de software”

Precondiciones

< Precondición Uno >

El curso debe estar abierto.

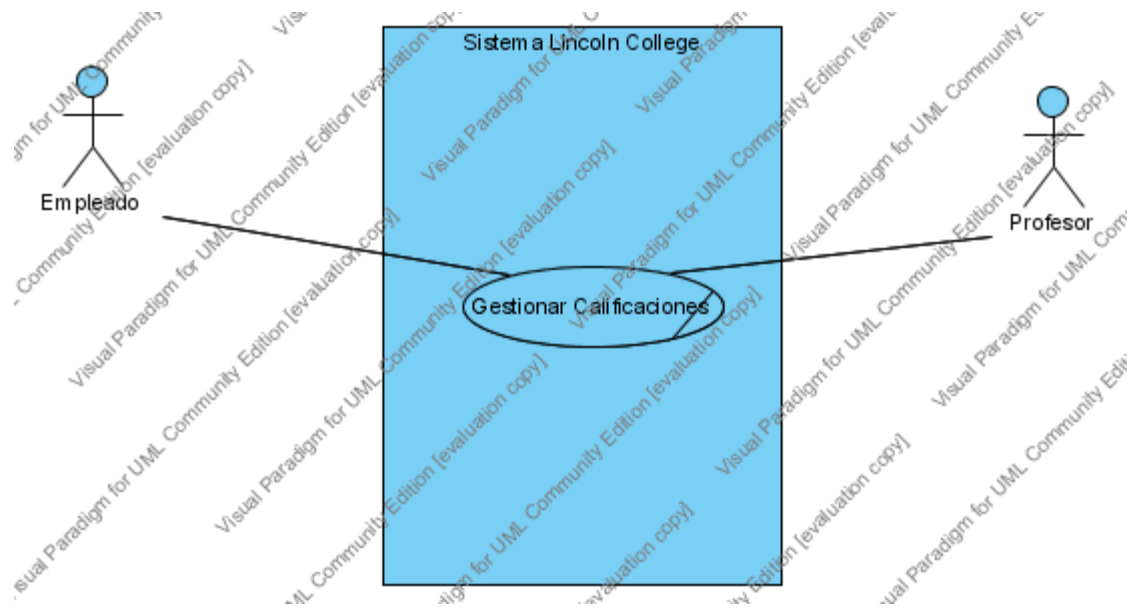
Poscondiciones

< pos condición Uno >

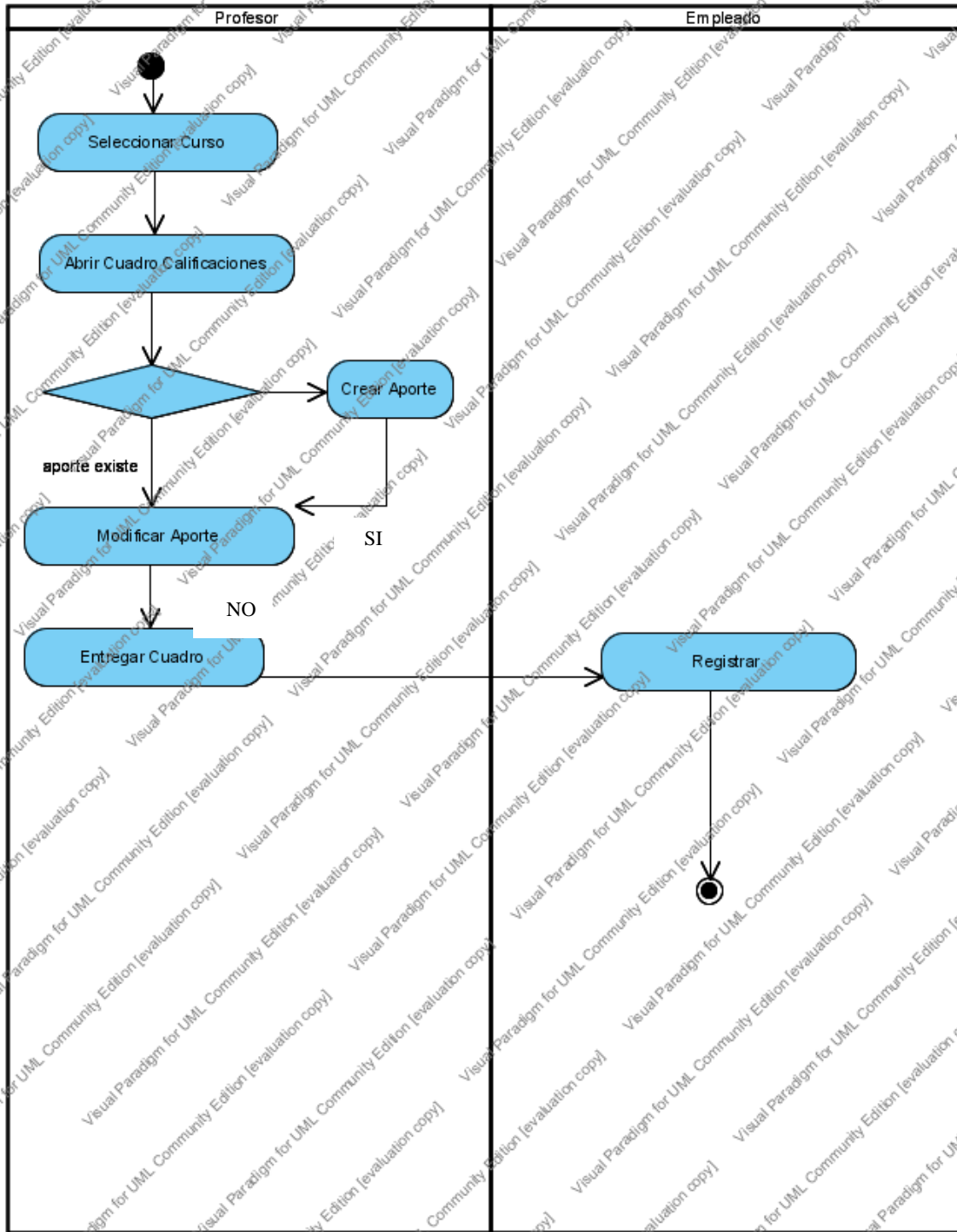
Los aportes han sido creados exitosamente y sus valores ingresados.

SECCIONES

Sección 11.1 Diagrama de Casos de Uso del Negocio



Sección 11.2 Diagrama de Actividades del Negocio



FASE: INICIO	ID: INI006
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCION:	Especificación del Caso de Uso del Negocio: Gestionar Asistencia
VERSION:	1.0

© 2010 Lincoln College of Foreign Languages
 TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.
 No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Especificación de Caso de Uso del Negocio:

Gestionar Asistencia

Nombre del Caso de Uso del Negocio

Breve Descripción

Este caso de uso describe las actividades que se lleva a cabo dentro del ámbito de gestión de asistencia de alumnos.

Metas

Tipo de Meta
Agilizar el proceso de registro de asistencia
Disponer de los datos de asistencia eficazmente.

Objetivos de Desempeño

Objetivo de desempeño 1

Agilizar el proceso de registro de asistencia.	10%	100%
--	-----	------

Objetivo de desempeño 2

Disponer de los datos de asistencia eficazmente.	20%	100%
--	-----	------

Flujo de Eventos

A continuación se realiza breve descripción del caso de uso, en la sección 11.2 se encuentra la descripción detallada mediante un diagrama de actividades.

Flujo Básico

El profesor inicia la clase

El profesor toma la asistencia de los alumnos del curso por cada estudiante presente
coloca una marca Ver FA1 , FA2

El profesor cierra la asistencia

El profesor registra los datos de la clase: fecha , hora inicio , hora finalización para
ingreso en la Secretaría.

El profesor termina la clase

El profesor entrega los datos al final del día en Secretaría

El empleado registra los datos

Finaliza el caso de uso.

Flujos Alternos

FA1. <El estudiante no se encuentra presente>

El profesor espera 15 minutos, si el alumno no llega, será considerado como ausente

FA2. <Ningún estudiante está presente >

Si no acude a clase ningún estudiante hasta 15 minutos después del inicio de la clase, el profesor cierra la asistencia y termina la clase

Reglas de Negocio

RN1. <Asistencia de estudiantes y profesores >

La asistencia es tomada por cada profesor diariamente al inicio de la clase como primera actividad.

El profesor puede incluir en la lista del día a un estudiante con un tiempo de retraso hasta 15 minutos.

Si el estudiante se pasa de ese tiempo es automáticamente ausente.

Si ningún estudiante llega a clase dentro de 15 minutos, se cierra la clase y se registra como clase dada.

Si el profesor no llega dentro de los 15 minutos, este debe recuperar si es posible dentro del mismo día, caso contrario se añade una clase al final del periodo.

Riesgos

El riesgo para este caso de uso viene de la posibilidad del olvido de los profesores de ingresar la asistencia de un periodo de clase.

Dueño del Proceso

El dueño del proceso de negocio es el profesor.

Requerimientos Especiales

Referirse al documento “Especificaciones suplementarias de software”

Precondiciones

< Precondición Uno >

El profesor debe estar asignado al curso en cuestión.

Los estudiantes deben estar matriculados al curso.

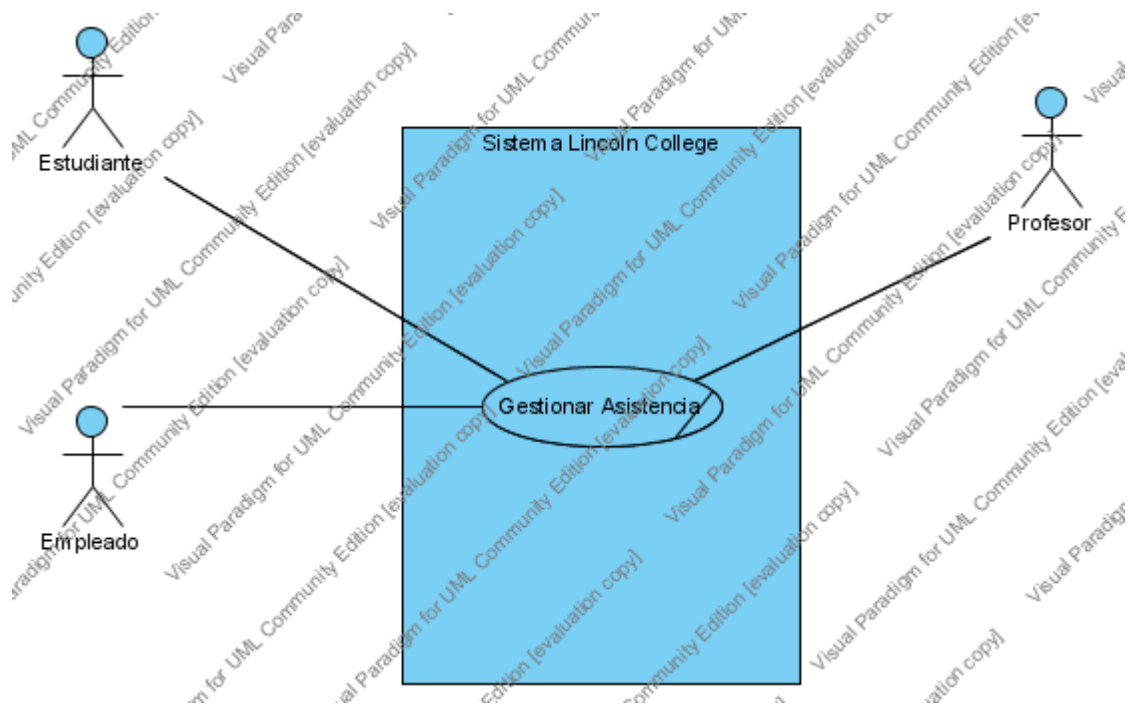
Poscondiciones

< pos condición Uno >

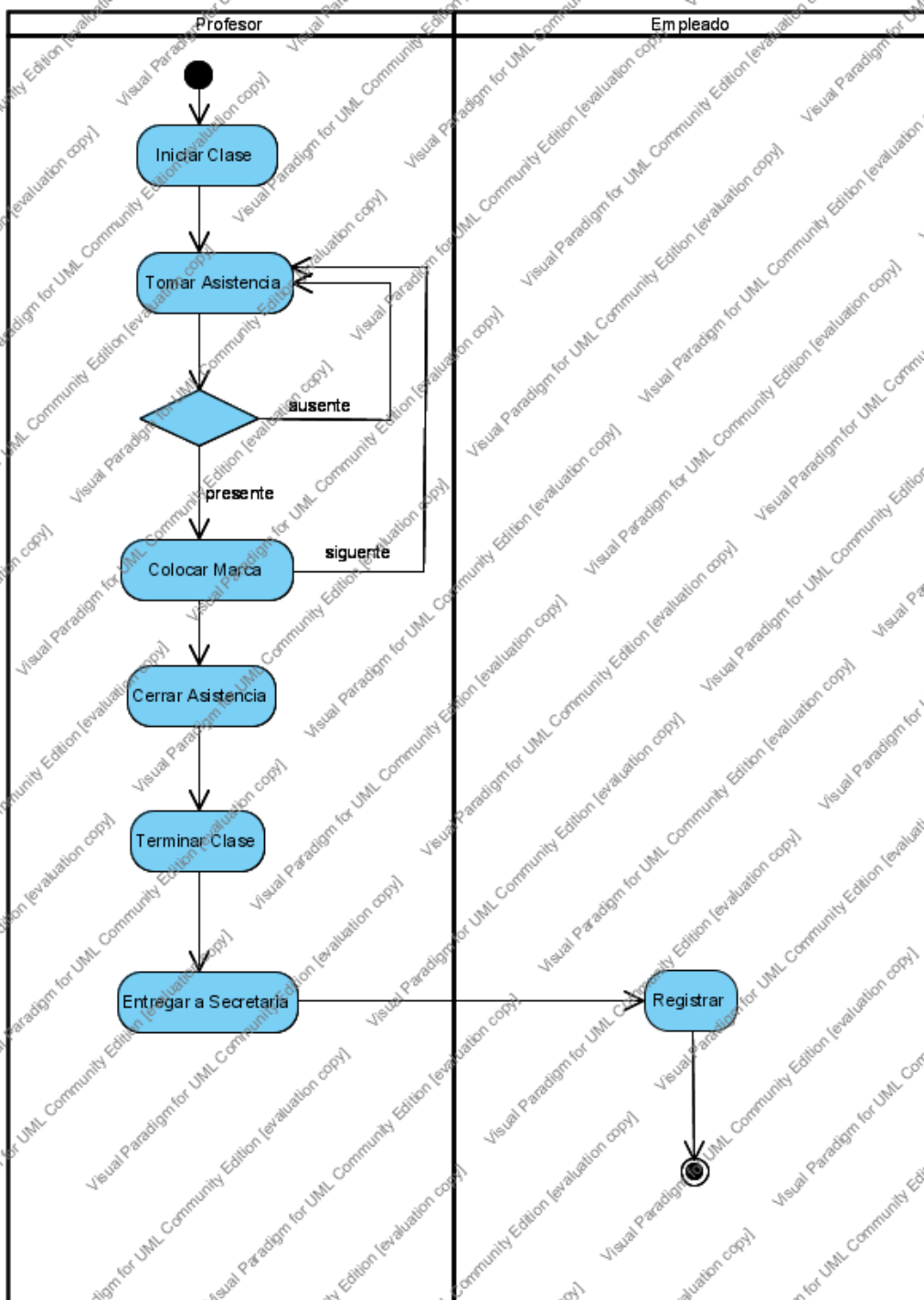
La asistencia se encuentra ingresada correctamente.

SECCIONES

Sección 11.1 Diagrama de Casos de Uso del Negocio



Sección 11.2 Diagrama de Actividades del Negocio



FASE: INICIO	ID: INI007
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCION:	Especificaciones Suplementarias de Software
VERSION:	1.0

© 2010 Dirección de Desarrollo Institucional del IESS
 TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

4.3.4 Especificaciones Suplementarias de Software

1. Funcionalidad

(Ver documento Documento Especificaciones de Casos de Uso del Negocio)

2. Usabilidad

- Las interfaces del sistema serán consistentes
- Se utilizará ayudas contextuales en los controles de las interfaces
- Los menús se presentará en forma de árbol con hijos (TreeMenu)
- Sistema orientado a la web con extensivo uso de Ajax , que es el objetivo de este proyecto.

3. Confiabilidad

El sistema estaría proyectado para uso durante las horas de oficina si fuera puesto en producción

El tiempo de subida de los servicios se estima en unos cinco minutos

Tecnología de servidor robusta.

Datos serán almacenados en un RDBMS seguro y con mecanismos probados de recuperación de datos y elaboración/recuperación de respaldos.

4. Desempeño

El tiempo de respuesta del sistema en una petición asíncrona va desde los milisegundos a 2 segundos.

El tiempo de respuesta del sistema a una petición síncrona es de máximo 5 segundos

El sistema soporta hasta 70 usuarios en línea concurrentes (firefox,google chrome) en la intranet.

Extjs (Ajax) a través de peticiones asíncronas permite una mejor utilización de la memoria, el ancho de banda lo que se traduce en una mejor experiencia para el usuario.

5. Soporte

No se incluye soporte ya que la aplicación no va a ser desplegada en producción.

6. Restricciones de diseño

El lenguaje a utilizar en el servidor será PHP 5.2.

La capa de persistencia de datos será Active Record.

La aplicación estará desarrollada con el framework Extjs Versión 3.1.

La arquitectura será MVC Model View Controller.

7. Documentación de usuario en línea y requerimientos de ayuda del sistema

- El sistema tendrá ayudas contextuales en línea.
- Existirá un manual de usuario.

8. Componentes comprados

- El sistema estará realizado con tecnologías open source.

9. Interfaces

- Se utilizará el protocolo TCP/IP para comunicaciones en la intranet.

10. Requerimientos de licenciamiento

Open source , MIT

11. Estándares aplicables

Web 2.0

FASE: INICIO	ID: INI008
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCION:	Lista de Riesgos
VERSION:	1.0

© 2010 Lincoln College of Foreign Languages
 TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.
 No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

4.3.5 Lista de Riesgos

1. Introducción

A fin de garantizar el cumplimiento total de los objetivos del proyecto, es menester identificar aquellos factores que podría incidir negativamente en él. Se describen los riesgos, su posible mitigación, así como las salvaguardas que permitirán reducir su impacto, en caso de que el riesgo se materialice.

2. Propósito

Definir las estrategias a seguir, a fin de mitigar la materialización de riesgos o reducir sus impactos.

3. Alcance

Como parte del proceso de automatización de procesos del negocio importantes del Instituto de Idiomas Lincoln College of Foreign Languages se ha identificado riesgos potencialmente perjudiciales para el presente proyecto y para la institución misma. Al intentar mitigar estos riesgos, se asegura que cualquier esfuerzo de mejora tenga el curso esperado.

4. Definiciones, Siglas y Abreviaturas

LICENCIA DE SOFTWARE. Documento que habilita a su propietario, la utilización de un aplicativo informático y que legaliza su uso.

5. Descripción

Cada uno de los riesgos identificados, serán explicados en las secciones subsiguientes.

6. Riesgos

6.1 Falta de presupuesto para el proyecto — R01

6.1.1 Descripción

Dentro del presupuesto de la Institución, se debe hacer una asignación para la modernización de ella. Esto viene a ser una inversión para la cual las empresas deben separar recursos anticipadamente. La falta de recursos económicos puede significar que no se realice el proyecto o que solo quede plasmado en documentos sin resultados tangibles. O que este esfuerzo quede a medias debido a una asignación insuficiente.

6.1.2 Indicadores

- Inversión que traslada o desvía fondos del presupuesto destinados a las actividades clave del negocio.
- Demora en la puesta en marcha de un proyecto de modernización por falta de presupuesto.
- La falta de disponibilidad de fondos para la adecuación con la infraestructura necesaria.

6.1.3 Impacto, Probabilidad y Magnitud

Esta información se coloca en la matriz de riesgos de la sección 3 de este documento.

6.1.4 Estrategia Para la Mitigación

- Planificación de la inversión y los gastos asociados en el Presupuesto Operativo sección Costos Fijos para la inversión y Costos variables para gastos asociados .
- Monitoreo y seguimiento estricto.

6.1.5 Plan de Contingencias

- Alianzas estratégicas que permitan apoyar el esfuerzo.
- Establecer un fondo que permita la inversión en modernización.
- Acceder a créditos concedidos a la pequeña industria

6.2 Falta de cooperacion del recurso humano para puesta en marcha del proyecto — R02

6.2.1 Descripción

Cualquier esfuerzo de cambio en una organización remueve las bases de esta y las partes afectadas deben adaptarse mediante capacitaciones, talleres y charlas. La resistencia al cambio es un fenómeno muy común en las empresas y lo ha sido a través de los años. Esto puede entorpecer un proyecto hasta llevarlo a tomar más tiempo del previsto e incluso a fracasar en sus objetivos y metas propuestas.

6.2.2 Indicadores

- Actividades del proyecto no se han sido dados la prioridad que amerita y no se les presta la debida atención.
- No existe cooperación dentro de la organización para el proyecto.
- Mala actitud de los entrevistados o funcionarios de la institución demostrando así resistencia al cambio.
- Comentarios o rumores de despidos de personal por parte de los empleados.
- Insatisfacción de los empleados con las condiciones actuales de trabajo y remuneración.

6.2.3 Impacto, Probabilidad y Magnitud

Esta información se coloca en la Matriz de Registro de Riesgos más adelante.

6.2.4 Estrategia Para la Mitigación

- Socialización del proyecto enfocándose en aquellos aspectos que pueden ser fuente de mala interpretación o mala información.
- Priorizar la crítica constructiva y la generación de ideas por parte del personal.
- Promover el trabajo en equipo.

6.2.5 Plan de Contingencias

- Reuniones con el personal para hacerles conocer las intenciones del proyecto y como este ayudará a sus tareas diarias.
- Establecer políticas de motivación para quienes colaboran con este esfuerzo.
- Dar al proyecto la debida atención en lo que se refiere a recursos y personal.

7. Matriz de Registro de Riesgos

La siguiente matriz resume la información relevante de los riesgos identificados. La forma de valoración de impacto, probabilidad y Magnitud se documenta en la “Sección Definiciones de Probabilidad e Impacto”

Tabla 4.11: Matriz de Riesgos para el proyecto.

Identificación		Valoración			Plan de Acción			
No. Riesgo	Riesgo	Impacto	Prob(%)	Prioridad (Magnitud)	Responsable	Disparador	Medición	Estatus
(1)	(2)	(4)	(5)	(6)	(7)	(8)	(9)	(12)
R01	Falta de presupuesto para el proyecto	0.8	0.5	0.6	Gerencia de la Institución	5 días	Semanal	Abierto
R02	Falta de cooperación del recurso humano para puesta en marcha del proyecto.	0.6	0.5	0.4	Recursos Humanos	15 días	Quincenal	Abierto

7.1 Definiciones de Probabilidad e Impacto

Para definir la matriz de probabilidad e impacto, se determinarán independientemente cada uno de los índices, dándole un valor numérico para cada uno de los ítems, en base a las siguientes tablas:

7.1.1 Probabilidad

Muy Probable	0.9
Bastante Probable	0.7
Probable	0.5
Poco Probable	0.3
Improbable	0.1

7.1.2 Impacto

Muy Alto	0.8
Alto	0.4
Moderado	0.2
Bajo	0.1
Muy Bajo	0.05

Para definir el nivel de impacto del riesgo en el proyecto se utilizará la siguiente tabla:

Tabla 4.12: Escalas de Impacto de un Riesgo sobre los Principales objetivos del Proyecto

Condiciones Definidas para Escalas de Impacto de un Riesgo sobre los Principales Objetivos del Proyecto					
<i>(Sólo se muestran ejemplos para impactos negativos)</i>					
Objetivo del Proyecto	Se muestran escalas relativas o numéricas				
	Muy bajo/0,05	Bajo/0,10	Moderado/0,20	Alto/0,40	Muy alto/0,80
Costo	Aumento de costo insignificante	Aumento del costo <10%	Aumento del costo del 10-20%	Aumento del costo del 20-40%	Aumento del costo del >40%
Tiempo	Aumento de tiempo insignificante	Aumento del tiempo <5%	Aumento del tiempo del 5-10%	Aumento del costo del 10-20%	Aumento del tiempo del >20%
Alcance	Disminución del alcance apenas perceptible	Áreas del alcance secundarias afectadas	Áreas de alcance principales afectadas	Reducción del alcance inaceptable para la CCSS	El elemento terminado del proyecto es efectivamente inservible
Calidad	Degradación de la calidad apenas perceptible	Sólo las aplicaciones muy exigentes se ven afectadas	La reducción de la calidad requiere la aprobación de la CCSS	Reducción de la calidad inaceptable para la CCSS	El elemento terminado del proyecto es efectivamente inservible

Una vez definida tanto la probabilidad como el impacto del proyecto, se calculará el índice P×I, el cual, se obtiene de la multiplicación de los índices individuales de probabilidad e impacto respectivamente y se asocia a alguna casilla de la siguiente matriz, la cual define, de forma cualitativa en nivel del riesgo.

Tabla 4.13 : Matriz de Probabilidad e Impacto

Probabilidad	Amenazas					Oportunidades				
0.90	0.05	0.09	0.18	0.36	0.72	0.72	0.36	0.18	0.09	0.05
0.70	0.04	0.07	0.14	0.28	0.56	0.56	0.28	0.14	0.07	0.04
0.50	0.03	0.05	0.10	0.20	0.40	0.40	0.20	0.10	0.05	0.03
0.30	0.02	0.03	0.06	0.12	0.24	0.24	0.12	0.06	0.03	0.02

Tabla 4.14 : Calificación del Riesgo basado en la matriz de Probabilidad

Riesgo	Impacto	Probabilidad	Indice	Tipo	Calificación Riesgo
R01	0.8 (Alto)	0.5 (Mediana)	0.40	Amenaza	Mediano
R02	0.6 (Medio Alto)	0.5 (Mediana)	0.30	Amenaza Oportunidad	Mediano

El Riesgo R01 es una Amenaza de Mediano Riesgo que podría ser transformado de Riesgo a Problema.

El Riesgo R02 es una Amenaza y una Oportunidad de Mediano Riesgo por lo que debería ser aprovechado como una Oportunidad y así mitigar este riesgo que podría convertirse de Amenaza en Problema si no se toma las acciones de motivación del Recurso Humano.

4.4 FASE DE ELABORACION

El objetivo primordial de la fase de Elaboración dentro de la Metodología RUP es la elección de una arquitectura candidata para el proyecto de software.

Esto se ve reflejado en el documento SAD (Software Architecture Document) dentro del cual se tienen las 4 vistas de la arquitectura del sistema.

Esta fase también incluye la realización del Plan de Elaboración que nos acerca a la fase de Construcción.

4.4.1 ESPECIFICACION DE CASOS DE USO

Se divisaron 10 casos de uso que incluyen casos principales y de apoyo entre los cuales tenemos:

1. Ingresar y Mantener Datos
2. Matricular Alumnos
3. Crear Curso
4. Asignar Profesor
5. Gestionar Calificaciones
6. Ingresar Asistencia
7. Mantener Parámetros Corporativos
8. Generar Reporte de Asistencia
9. Generar Reportes y Estadísticas
10. Generar Rol de pagos de Profesores

FASE: ELABORACIÓN	ID: ELB001
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de uso: Ingresar y Mantener Datos
VERSIÓN:	1.0

2010 Lincoln College of Foreign Languages
 TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Especificación de Caso de Uso : Ingresar y Mantener Datos

Breve Descripción

El caso de uso permite al usuario ingresar y mantener los datos de personas que incluye datos personales, contactos, datos de empleado y datos de estudiante.

Flujo de Eventos

Flujo Básico

1. El caso de uso empieza cuando el usuario operador selecciona la opción Ingreso y Mantenimiento de Datos.
2. El operador ingresa la cédula de la persona.
3. El sistema determinará si la persona existe en el sistema. Si no existe ver FA2
4. Si la persona existe en el sistema, se poblan los campos de la sección de datos personales, contactos, profesores, matriculas actuales.

Los datos se despliegan de acuerdo a lo siguiente:

Datos personales

1. Cédula
2. Nombres
3. Apellidos
4. Fecha de Nacimiento
5. Sexo
6. Estado Civil
7. Nacionalidad

Dirección

1. Calle
2. Transversal
3. Piso
4. Número
5. Urbanización
6. Sector
7. Referencia
8. Ciudad

9. País

10. Sitio de la Dirección

Contactos

1. Sitio

2. Tipo Contacto

3. Valor del contacto

4. Fecha Ingreso del contacto

5. Fecha de Modificación del contacto

2. Si la persona que si existe en el sistema es un empleado se poblan las secciones de idiomas, disponibilidad e información salarial.
3. El operador puede proceder a modificar los datos.
4. Una vez realizados los cambios necesarios a los datos , se presiona el botón Guardar
5. El sistema almacena o modifica los datos ingresados.
6. Finaliza el caso de uso.

Flujos Alternos

FA2. La persona no existe en la base de datos

1. El usuario ingresa el número de cédula de la persona
2. El sistema no despliega ningún dato ya que la persona no existe.
3. El usuario ingresa los siguientes datos:
 - **Datos personales**
 - Cedula o pasaporte de la persona.
 - Nombres: Los dos nombres de la persona
 - Apellidos: Los dos apellidos de la persona
 - Fecha de Nacimiento: La fecha de nacimiento de la persona desplegada en un control tipo calendario
 - Sexo: Masculino o Femenino seleccionado de un grupo de botones radio.
 - Estado Civil: Estado civil puede ser seleccionado de una caja combo y puede ser uno de los siguientes valores:
 - SOLTERO
 - CASADO
 - VIUDO
 - UNION LIBRE
 - DIVORCIADO

➤ Nacionalidad : El usuario debe seleccionar una nacionalidad de un control combo.

- **Dirección**

➤ Calle: El nombre de la calle principal de la dirección.

➤ Transversal: El nombre de la transversal más próxima a la dirección.

➤ Piso: El número de piso donde se ubica la persona en la dirección.

➤ Numero: El número o código del inmueble en mención.

➤ Urbanización: El nombre de la urbanización sin comillas.

➤ Sector: El nombre del sector de la dirección.

➤ Referencia: Datos de referencia de la dirección que ayudarán a su ubicación.

➤ Ciudad: Ciudad en donde está ubicada la dirección.

➤ País: El nombre del país donde está ubicada la dirección.

➤ **Sitio de la Dirección:** El que puede uno de los siguientes valores.

- CASA

- TRABAJO

- ESCUELA

- UNIVERSIDAD

- CASA CAMPO

- AVION PRIVADO
 - CASA EN EL EXTRANJERO
 - CASA VACACIONAL
 - MOVIL
- **Contactos** : En una tabla se ingresan uno o varios contactos con los siguientes datos.
 - **Sitio** : El cual es seleccionado de una caja combo y puede tomar uno de los siguientes valores:
 - CASA
 - TRABAJO
 - ESCUELA
 - UNIVERSIDAD
 - CASA CAMPO
 - AVION PRIVADO
 - CASA EN EL EXTRANJERO
 - CASA VACACIONAL
 - MOVIL

- **Tipo Contacto:** El cual es seleccionado de una caja combo de selección y puede tomar uno de los siguientes valores:
 - TELEFONO
 - CELULAR
 - FAX
 - BEEPER
 - PAGER
 - EMAIL
 - WEBSITE

- **Valor del contacto:** El valor del contacto ingresado por teclado que puede contener todo tipo de caracteres según el tipo.

- **Fecha Ingreso del contacto:** Fecha de ingreso del contacto al sistema
(AUTOMATICO POR SISTEMA)

- **Fecha de Modificación del contacto:** Fecha de modificación de datos del contacto
(AUTOMATICO POR EL SISTEMA)

Precondiciones

1. El usuario debe tener un rol autorizado para mantenimiento de datos del sistema
2. La persona debe proporcionar los datos requeridos.

Poscondiciones

1. La persona ingresada al sistema correctamente.
2. Los datos de la persona han sido modificados exitosamente.

FASE: ELABORACIÓN	ID: ELB002
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de uso: Matricular Alumnos
VERSIÓN:	1.0

TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Especificación de Caso de Uso

Matricular Alumnos

Breve Descripción

El caso de uso permite al usuario matricular uno o varios alumnos en un curso creado y asignado.

Flujo de Eventos

Flujo Básico

1. El usuario operador selecciona la opción Nueva Matricula
2. El sistema despliega la ventana de creación de matricula con los siguientes campos:
 - Id de Matricula
 - Id del Alumno
 - Cédula
 - Nombres
 - Apellidos
3. El operador ingresa la cedula del alumno a matricular

4. El sistema despliega los siguientes datos Ver FA1
 - Id de Matricula
 - Id del Alumno
 - Cedula
 - Nombres
 - Apellidos
5. El operador presiona el botón Ingresar Matricula.
6. El sistema despliega el resultado del ingreso de la matricula
7. Si se va a ingresar más alumnos se vuelve al paso 1.
8. El caso de uso finaliza.

Flujos Alternos

FA1. El alumno no existe en el sistema

1. El usuario operador ingresa la cedula , nombres y apellidos del alumno
2. El usuario operador presiona el botón Crear Persona
3. El sistema habilita el botón Crear Alumno
4. El sistema despliega ventada de resultado de creación de estudiante
5. Continúa en el flujo básico ítem 5

Precondiciones

1. El usuario debe tener un rol autorizado
2. Los casos de uso Crear Curso y Asignar Profesor deben haberse ejecutado exitosamente.

Poscondiciones

1. El / los alumno/s han sido matriculados exitosamente.

FASE: ELABORACIÓN	ID: ELB003
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de uso: Crear Curso
VERSIÓN:	1.0

TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Especificación de Caso de Uso:

Crear Curso

Breve Descripción

El caso de uso permite al usuario crear cursos a través del proceso de matriculación.

Flujo de Eventos

Flujo Básico

1. El usuario operador selecciona la opción Proceso de Matriculación.
2. El operador presiona el botón Configuración para ingresar una configuración de curso a crear.
3. El sistema despliega las opciones para la configuración del curso que incluyen:
 - Numero de nivel
 - Idioma del curso
 - Modalidad (Individual , Grupal)
 - Tipo de nivel (Normal, Intensivo, Fines de semana)
 - Modalidad de horario (Horas, Completo)
 - Número de horas en caso se escoja la opción Modalidad Horario Horas.

4. El operador presiona el botón Obtener precio
5. El sistema consulta los datos del nivel ingresados/seleccionados por el usuario y despliega el Nivel Id con el precio respectivo del nivel.
6. El operador presiona el botón Guardar. Ver FA2
7. El sistema despliega una pantalla de confirmación
8. El usuario acepta la confirmación presionando el botón Si. Ver FA3
9. El sistema almacena y despliega los datos del curso en el formulario principal.
10. El operador selecciona las fechas de inicio y finalización del curso
11. El operador presiona el botón Crear.
12. El sistema despliega una pantalla de confirmación
13. El usuario acepta la confirmación. Ver FA3
14. El sistema almacena los datos del curso y despliega una ventana de confirmación de la creación del curso.
15. El sistema despliega el id de curso creado.
16. El sistema habilita las opciones para la asignación de profesor y horario al curso creado.

17. El caso de uso finaliza.

18. Finaliza el caso de uso.

Flujos Alternos

FA2. El usuario elige cancelar o resetear

Si el usuario elige cancelar, se cierra la ventana

Si el usuario elige resetear, se limpian los campos del formulario para seleccionar de nuevo los datos del nivel

FA3 . El usuario decide no guardar el curso

El usuario presiona el botón No para rechazar la confirmación.

Se cierra la venta de confirmación.

El curso no es creado.

Precondiciones

1. El usuario debe tener un rol autorizado para mantenimiento de datos del sistema
2. Los datos del nivel deben estar creados con anterioridad en el sistema.

Poscondiciones

1. El curso ha sido creado correctamente.

FASE: ELABORACIÓN	ID: ELB004
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de uso: Gestionar Calificaciones
VERSIÓN:	1.0

© TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Especificación de Caso de Uso : Gestionar Calificaciones

Breve Descripción

El caso de uso permite al usuario ingresar y mantener calificaciones de los cursos a él asignados.

Flujo de Eventos

Flujo Básico

1. El usuario con rol de profesor selecciona la opción Ingreso de calificaciones
2. El profesor selecciona la opción Buscar Curso.
3. El sistema despliega un cuadro con los cursos activos del profesor.
4. El usuario selecciona el curso cuyo cuadro de aportes desea ver .
5. El usuario presiona el botón Seleccionar Curso.
6. Para ingresar un nuevo aporte FA1, el profesor selecciona la opción Crear Aporte , en donde debe ingresar los siguientes valores:
 - Id del curso
 - Tipo de aporte a crearse

El nivel puede tener máximo dos tipos de cada aporte , un examen final y un proyecto.

Los tipos de aportes disponibles son:

D=DEBERES

C=COMPOSICION

PE= PRUEBA ESCRITA

PO= PRUEBA ORAL

PR= PROYECTO

EX= EXAMEN DE FINAL DE PERIODO

EE= EXAMEN ESCRITO

EO= EXAMEN ORAL

7. El sistema despliega inmediatamente el nuevo aporte insertado
8. El usuario ingresa la calificación del estudiante
9. El sistema guarda la calificación ingresada
10. Finaliza el caso de uso.

Flujos Alternos

FA1. El usuario va a actualizar una calificación

1. El usuario seleccionar un curso de la lista.
2. El sistema despliega un cuadro de aportes
3. El usuario hace doble clic en el cuadro donde desea modificar e ingresa la nueva calificación:

4. El sistema recalcula los totales inmediatamente y los despliega.
5. El caso de uso termina

FA2. El usuario cierra el curso

1. El usuario selecciona un curso de la lista.
2. El usuario se asegura de que todo esté listo para cerrar el curso
3. El usuario selecciona la opción Cerrar Curso
4. El sistema pide una confirmación para realizar el cierre
5. El usuario confirma o cancela la acción
6. Si el usuario confirma presionando el botón Si
7. El sistema cierra el curso y ya no puede ser editadas las calificaciones del mismo
8. Una vez cerrado el curso , termina el caso de uso

Precondiciones

1. El usuario debe tener un rol autorizado de profesor
2. Debe existir cursos abiertos a nombre del profesor

Poscondiciones

Las calificaciones han sido creadas con éxito

Las calificaciones han sido modificadas con éxito

El curso se encuentra cerrado.

FASE: ELABORACIÓN	ID: ELB005
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de uso: Ingresar Asistencia
VERSIÓN:	1.0

TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Especificación de Caso de Uso del Negocio : Ingresar Asistencia

Breve Descripción

El caso de uso permite el ingreso de la asistencia de los estudiantes.

Flujo de Eventos

Flujo Básico

1. El usuario selecciona la opción Operaciones Curso.
2. El sistema despliega los cursos a cargo del usuario.
3. El usuario elige el curso de su elección.
4. El sistema despliega los detalles del curso.
5. El usuario escoge la opción Ingresar Asistencia.
6. El sistema despliega la ventana de Ingreso de Asistencia.
7. El usuario inicia la clase.

8. El usuario escoge el horario del curso y los estudiantes presentes.
9. El usuario ingresa la asistencia.
10. El sistema ingresa la hora de profesor y la asistencia de los estudiantes.
11. El caso de uso termina.

Precondiciones

1. El usuario debe tener un rol autorizado para generar rol de pagos
2. Los estudiantes deben estar matriculados y asignados al curso .
3. El curso no debe estar cerrado.

Poscondiciones

La asistencia se ingreso correctamente.

FASE: ELABORACIÓN	ID: ELB006
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de uso: Mantener Parámetros Corporativos
VERSIÓN:	1.0

TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Especificación de Caso de Uso:

Mantener Parámetros Corporativos

Breve Descripción

El caso de uso permite al usuario ingresar y mantener los datos de los diferentes parámetros corporativos que maneja el instituto.

Flujo de Eventos

Flujo Básico

1. El caso de uso inicia cuando el usuario Administrador accede a la pantalla de Administración.
2. El sistema despliega los diferentes parámetros que pueden ser administrados.
3. El usuario Administrador escoge un parámetro.
4. El sistema despliega la pantalla del parámetro.
5. El usuario ingresa uno o más parámetros nuevos, modifica o elimina uno o más parámetros.

6. El sistema almacena los cambios realizados.

7. El sistema refresca las vistas utilizadas.

8. Finaliza el caso de uso.

Precondiciones

3. El usuario debe tener un rol autorizado como Administrador

Poscondiciones

El/Los parámetros corporativos han sido ingresados o modificados.

FASE: ELABORACIÓN	ID: ELB007
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de uso: Generar Reporte de Asistencia
VERSIÓN:	1.0

TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Especificación de Caso de Uso : Generar Reporte de Asistencia

Nombre del Caso de Uso del Negocio

Breve Descripción

El caso de uso permite la generación de Reportes de Asistencia por mes.

Flujo de Eventos

Flujo Básico

1. El usuario selecciona la opción Reporte de Asistencia
2. El usuario ingresa el año y selecciona el mes de la lista.
3. El usuario presiona el botón Generar Reporte.
4. El sistema genera el reporte.
5. El sistema guarda el reporte en el sistema de archivos del servidor.
6. El sistema despliega una ventana de resultado del proceso.

Precondiciones

4. El usuario debe tener un rol autorizado para generar reportes de asistencia

Poscondiciones

1. El reporte de asistencia se encuentra generado en el sistema de archivos del servidor web.

FASE: ELABORACIÓN	ID: ELB008
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de uso: Generar Reportes y Estadísticas
VERSIÓN:	1.0

TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Especificación de Caso de Uso :

Generar Reportes y Estadísticas

Breve Descripción

El caso de uso permite al usuario generar reportes y estadísticas.

Flujo de Eventos

Flujo Básico

1. El usuario administrador selecciona la opción Reportes.
2. El usuario escoge el reporte de su elección.
3. El usuario ejecuta el reporte seleccionado
4. Finaliza el caso de uso.

Flujos Alternos

FA1. El usuario es Operador

1. El usuario escoge la opción Reportes Asistencia
2. El sistema despliega los reportes disponibles
3. El usuario ingresa los parámetros para el reporte
4. El usuario ejecuta el reporte
5. El sistema despliega el reporte
6. El usuario imprime el reporte

Precondiciones

El usuario debe tener un rol autorizado para generar reportes y estadísticas

Poscondiciones

El/los reportes seleccionados se ejecutaron exitosamente.

FASE: ELABORACIÓN	ID: ELB009
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de uso: Generar Rol de Pagos
VERSIÓN:	1.0

TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

Especificación de Caso de Uso: Generar Rol de Pagos

Breve Descripción

El caso de uso permite la generación de Roles de Pago de Profesores de acuerdo a periodo mensual.

Flujo de Eventos

Flujo Básico

1. El usuario selecciona la opción Rol de Pagos.
2. El usuario ingresa el año y selecciona el mes de la lista.
3. El usuario presiona el botón Generar Rol.
4. El sistema genera el reporte.
5. El sistema despliega los datos del reporte.
6. El usuario escoge la opción Imprimir.

Precondiciones

El usuario debe tener un rol autorizado para generar rol de pagos

Poscondiciones

El rol de horas de profesores ha sido generado.

FASE: ELABORACIÓN	ID: ELB010
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Documento de Arquitectura SAD
VERSIÓN:	1.0

© Lincoln College of Foreign Languages
 TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio. No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

4.4.2 SAD (Documento de Arquitectura de Software)

Éste documento define la arquitectura global del sistema Escolástico Lincoln College.

A continuación se presentan las actividades a realizarse y las secciones de éste documento al cual está relacionada.

1. Propósito

El Documento de Arquitectura proporciona una descripción general del presente sistema. Éste presenta un número de vistas arquitectónicas que describen los diferentes aspectos del sistema.

Éste modelo permite que los roles busquen lo que necesitan en la arquitectura de software. En éste caso hablamos de:

1. Líderes de proyecto, quienes podrán determinar los recursos necesarios para tomar decisiones futuras de mantenimiento.
2. Analistas de sistemas, quienes podrán describir los casos de uso ó procesos de negocio que apoya el sistema.
3. Arquitectos de Software, quienes deben realizar el seguimiento y versionamiento del documento a medida que el proyecto incremente su funcionalidad y aparezcan nuevos mecanismos estándares que simplifiquen el desarrollo y la comprensión de futuros proyectos.
4. Desarrolladores, a quienes sirve como marco de referencia y protocolo para el desarrollo del proyecto.
5. Operadores de Servidores de Aplicaciones, Base de Datos y Redes, que les permita identificar los recursos del sistema del cual son custodios.

2. Alcance

El alcance de éste Documento de Arquitectura de Sistema es describir la arquitectura del Sistema Escolástico Lincoln College

3. Definiciones, acrónimos, y abreviaturas

RUP: Proceso Unificado de Racional

UML: Lenguaje Unificado de Modelado

SAD: Documento de Arquitectura de Sistema.

JSON: Javascript Object Notation.

4. Referencias

Tabla 4.16: Referencias para la arquitectura.

[KRU41]	The “4+1” view model of software architecture,	Philippe Kruchten, November 1995, http://www3.software.ibm.com/ibmdl/pub/software/rational/web/witepapers/2003/Pbk4p1.pdf
[RSA]	IBM Rational Software Architect	http://www-306.ibm.com/software/awdtools/architect/swarchitect/index.html
[RUP]	The IBM Rational Unified Process.	http://www-306.ibm.com/software/awdtools/rup/index.html
[RUPRSA]	Developing a J2EE Architecture with Rational Software Architect using the Rational Unified Process®, IBM DeveloperWorks	Jean-Louis Maréchaux, Mars 2005, http://www-128.ibm.com/developerworks/rational/library/05/0816_Louis/
	Extjs in Action.	Tony Garcia. http://java.sun.com/docs/books/jls/second_edition/html/names.doc.html
	Arquitectura, patrones, integración.	http://www.theserverside.com/

5. Resumen

El documento de arquitectura está estructurado de tal forma que facilite la comprensión por parte de los involucrados al momento de tomar decisiones de desarrollo y puesta en producción.

Con el fin de documentar todos los aspectos de la arquitectura, el documento de Arquitectura de Software contiene las siguientes subsecciones:

- Sección 2: Describe el uso de cada vista.
- Sección 3: Describe las restricciones arquitectónicas del sistema.
- Sección 4: Describe los requerimientos funcionales con un impacto significativo en la arquitectura.
- Sección 5: Describe la arquitectura desde un punto de vista lógico.
- Sección 6: Describe cómo el sistema debe ser puesto en producción. Contiene el Modelo de Producción.
- Sección 7: Describe la implementación y los subsistemas de la aplicación o con los que podría interactuar.
- Sección 8: Describe cualquier elemento persistente significativo. Contiene el modelo de datos.
- Sección 9: Describe cualquier aspecto relativo a los atributos de la calidad del servicio (QoS).

Sección 2: Representación Arquitectónica

Ésta sección detalla la arquitectura usando las vistas definidas en el modelo “4+1”. Las vistas a ser empleadas en la descripción de éste documento son:

Tabla 4.17: Representaciones Arquitectónicas del Documento SAD

Vista	Audiencia	Área	Artefactos Relacionados
Proceso	Integradores	Requerimientos No Funcionales: Describe aspectos de diseño tales como concurrencia y sincronización.	
Implementación	Programadores	Componentes de Software: Describe las capas y subsistemas de la aplicación	Modelo de Implementación, Componentes
Producción	Administradores de Producción	Topología: Describe el mapeo del software en el hardware y muestra la distribución física del sistema.	Modelo de Producción
Casos de Uso	Todos los patrocinadores del sistema	Describe el conjunto de escenarios y casos de uso significativos que representan la funcionalidad principal del sistema.	Modelo de Casos de Uso, Documentos de Casos de Uso.
Datos	Administradores de Bases de Datos	Persistencia: Describe los elementos de la persistencia arquitectónicamente importantes en el modelo de datos.	Modelo de Datos.

Sección 3: Metas y restricciones de la arquitectura.

Ésta sección describe los requerimientos de software y los objetivos que tienen algún impacto significativo en la arquitectura.

1. Plataforma Técnica:

El sistema y todos los componentes que la integran esta desplegado en un servidor Web Apache 2.x

2. Transaccionalidad:

El sistema es transaccional aprovechando las capacidades de Active Record

3. Seguridad:

La aplicación debe implementar los siguientes comportamientos básicos de seguridad:

- Autenticación: Mediante una página de login con user y clave.
- Autorización: Acorde a su rol, se otorga a la ejecución de algunas acciones específicas registradas en las opciones de menú.
- Confidencialidad: La información sensible debe ser encriptada (cédula, clave).
- Integridad de Datos: Los datos enviados a través de la red no pueden ser interceptados y modificados.

4. Persistencia:

La persistencia de datos será orientada a una base de datos relacional Mysql versión 5 con una capa de persistencia de Active Record

5. Confiabilidad/Disponibilidad (tolerancia a fallas):

La disponibilidad de un sistema es un requerimiento clave por naturaleza, dado que es un sistema de reportes bajo demanda. La arquitectura candidata debe asegurar la capacidad de tolerancia a fallos.

La confiabilidad/disponibilidad será abordada por la plataforma.

La disponibilidad dispuesta es de 24/7: 24 horas al día, 7 días a la semana.

6. Desempeño:

El tiempo de respuesta de la aplicación debe ser menor igual a 10 segundos.

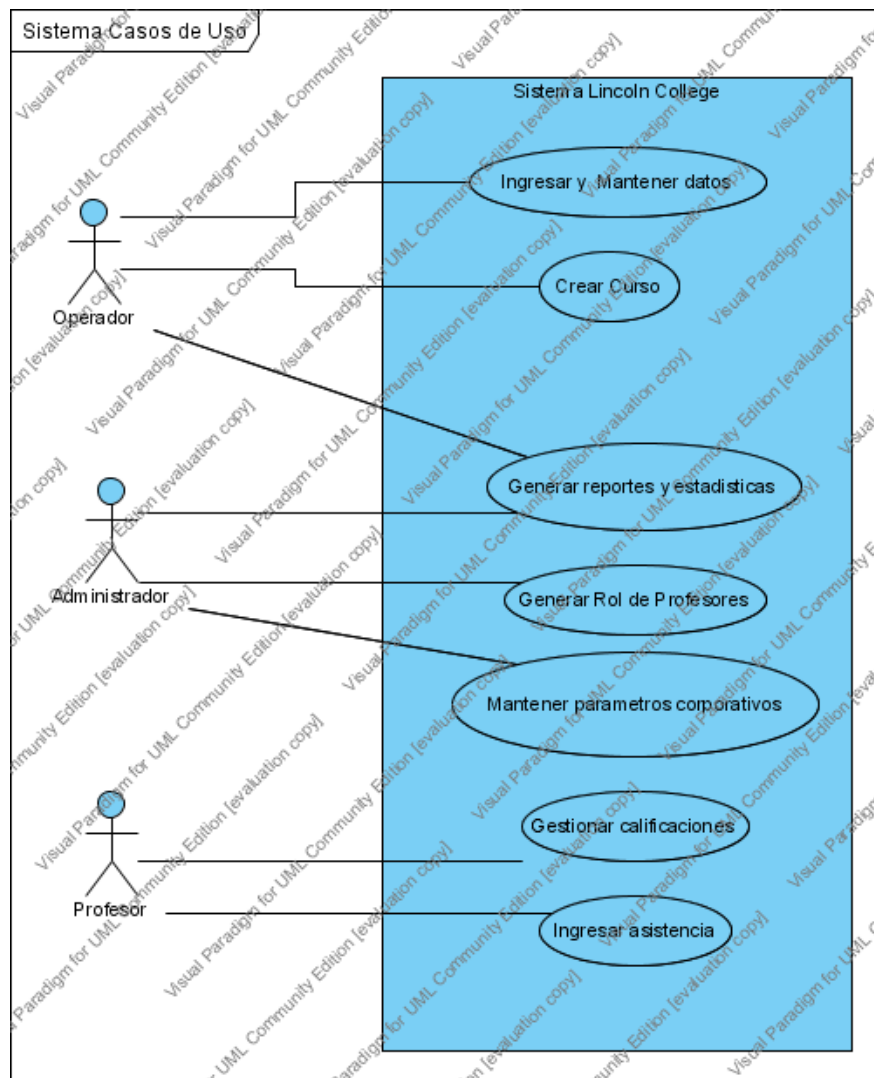
7. Internacionalización (i18n):

Dado que la aplicación estará disponible sólo a nivel nacional, no se requiere internacionalización.

Sección 3: Vista de casos de uso

Ésta sección lista los casos de uso o escenarios a partir del modelo de casos de uso si ellos representan la funcionalidad principal del sistema final.

Casos de uso relevantes a la arquitectura del sistema



Sección 4: Vista lógica de acuerdo a arquitectura en capas

1. Descripción General

El Sistema está dividido en capas, según la arquitectura N-capas que se muestra a continuación.

Arquitectura N-capas

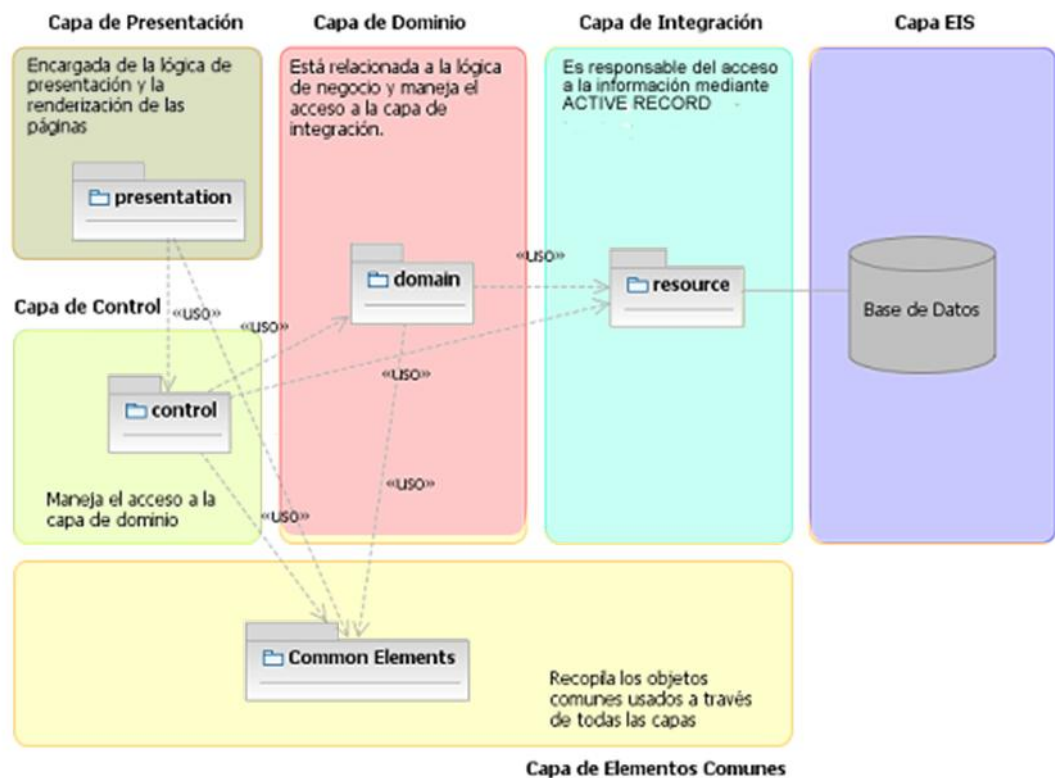


Gráfico 3.1 : Capas de la Arquitectura MVC del proyecto

El modelo de capas del Sistema Escolástico Lincoln College está basado en una estrategia de responsabilidad por capas. Ésta estrategia aísla varias responsabilidades del sistema una de otra, lo que mejora el desarrollo y el mantenimiento del mismo.

Cada capa tiene responsabilidades específicas.

- La capa de presentación se encarga de la lógica de presentación y la renderización de las páginas.
- La capa de control maneja el acceso a la capa de dominio.

- La capa de dominio está relacionada a la lógica de negocio y maneja el acceso a la capa de integración.
- La capa de integración (ActiveRecord) es responsable del acceso a la información institucional (Bases de Datos o en esta arquitectura llamado EIS).
- La capa de elementos comunes (incluidos en este proyecto)
- La capa EIS (Enterprise Information System) o Sistema de Información Empresarial donde persisten los datos de la aplicación. Contiene bases de datos relacionales, bases de datos orientadas a objetos, y sistemas legacy o antiguos.

Sección 5: Vista de Procesos

1. Procesos en Tiempo de Ejecución

Existe un solo proceso a tomar en cuenta, ya que el servidor Apache maneja hilos o threads para servir una petición por cada hilo.

2. Procesos de Carga de Datos

El sistema puede implementar cargas de datos en masa según las necesidades futuras del negocio en términos de ampliación de operaciones y datos. Al momento no es requerida dicha funcionalidad.

Sección 6: Vista de Despliegue

1. Descripción Principal

A continuación se presenta la estructura resumida de despliegue del sistema

Escolástico Lincoln College

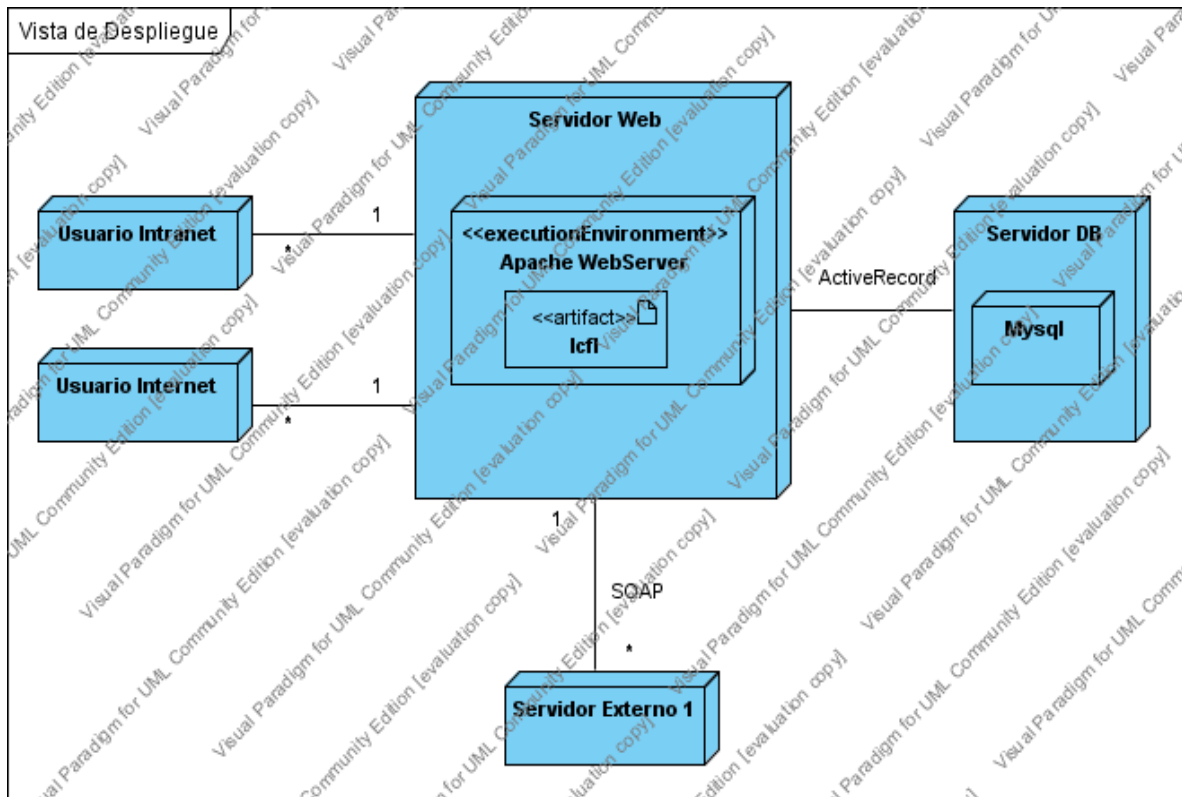


Gráfico 3.2: Vista de Despliegue del Sistema

El ambiente de despliegue se da dentro del servidor Web Apache donde se encuentra la aplicación lcfI. La aplicación recibe peticiones desde la intranet a través de direcciones de red locales, desde el internet por direcciones externas o mediante el consumo de un servicio web expuesto por parte de la aplicación accedido por medio del protocolo SOAP.

Una vez recibidas las peticiones, el servidor las enruta, procesa y devuelve una respuesta que puede ser JSON, HTML o XML dependiendo del tipo de acceso. Las peticiones a la base de datos las realiza la capa de persistencia manejada por Active Record.

Para el presente proyecto, solo se implementa los accesos por intranet.

Sección 7: Vista de Implementación

1. Descripción General

La Vista de Implementación describe la composición física de la implementación en términos de subsistemas, y otros elementos (directorios y archivos, incluyendo código fuente, datos, y archivos ejecutables).

Usualmente, las capas de la Vista de Implementación se ajustan a las capas definidas en la Vista Lógica.

2. Componentes que conforman el sistema

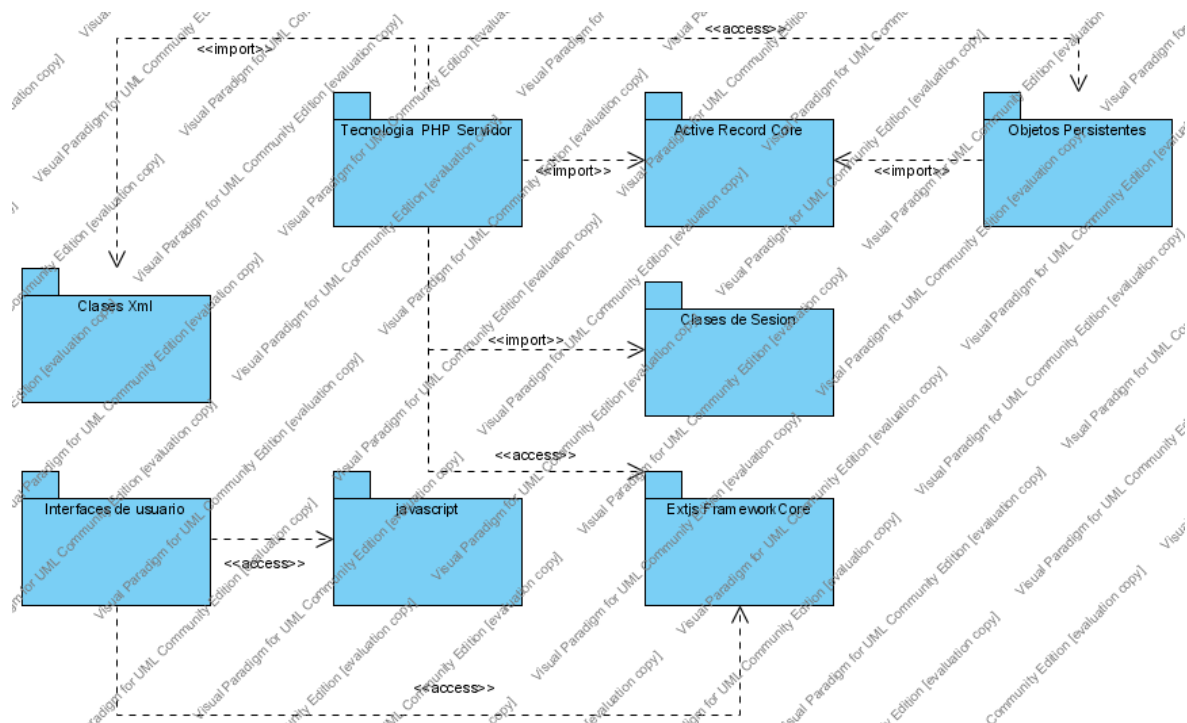


Gráfico 3.3: Componentes del sistema

En el gráfico anterior, se puede observar que las interfaces de usuario hacen uso de objetos y funcionalidad de javascript. Las interfaces de usuario al igual que el servidor interactúan con el core de Extjs. El método de transporte de datos puede ser JSON o XML.

A su vez en el lado del servidor, este accede a objetos persistentes a través de Active Record además de clases de sesión y clases para codificación y decodificación xml .

Sistemas con los que se puede interactuar

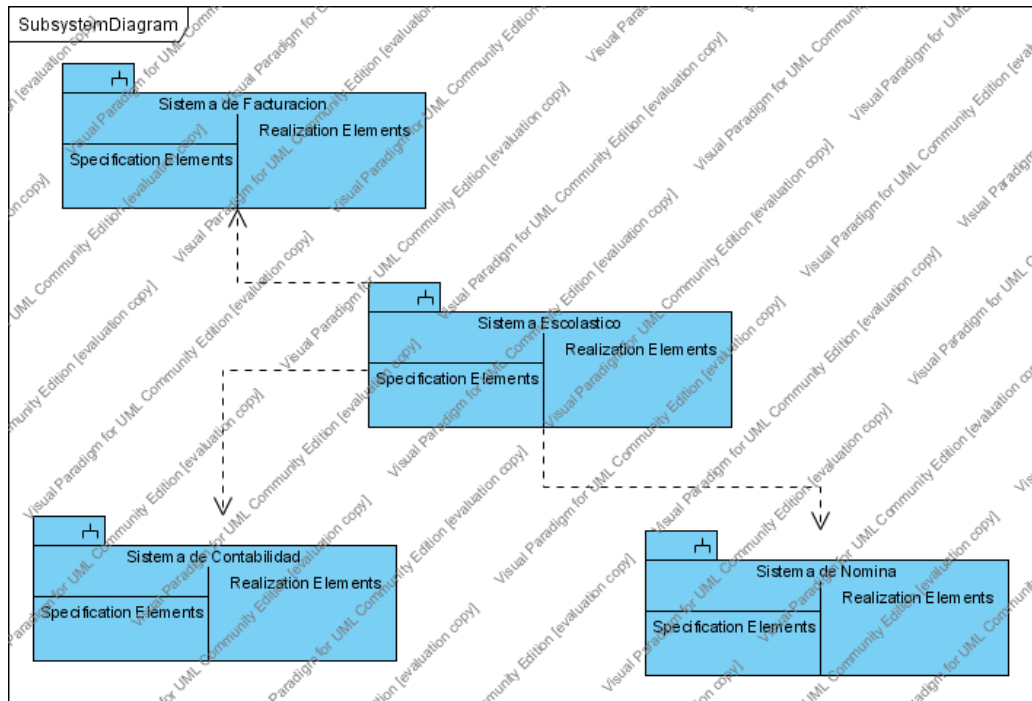


Gráfico 3.4: Diagrama de Subsistemas

Sección 8: Vista de Datos

(ANEXO B): DIAGRAMA ERD

Sección 9: Aspectos relativos a los atributos de la calidad del servicio (Qos)

1. Tamaño y Rendimiento

- El sistema soporta 100 usuarios en simultáneo contra el servidor Web.
- El sistema provee una respuesta menor a 10 segundos.

2. Calidad

- **Portabilidad:** El sistema permite su ejecución en cualquier sistema operativo.
Para su acceso necesita un explorador Web.
- **Mantenimiento:** El sistema, por estar separado en capas, provee facilidad para el incremento o modificación de funcionalidades.
- **Usabilidad:** El sistema se encuentra adaptado para cada usuario, permitiéndole acceder fácilmente a las funcionalidades que necesita, de manera rápida.
- **Funcionalidad:** El sistema provee todas las funcionalidades para las que fue concebido.

FASE: ELABORACIÓN	ID: ELB011
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Plan de Elaboración
VERSIÓN:	1.0

© 2009 Lincoln College of Foreign Languages
 TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio. No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.

4.4.3 Plan de Elaboración

1. Requerimientos Funcionales

Lista de los requerimientos funcionales revisados y descritos desde la primera revisión de los requerimientos candidatos. Se detalla los requerimientos a continuación:

Autenticación y sesiones.

Administración de parámetros corporativos.

Administración de usuarios del sistema

1. Creación y mantenimiento de perfiles de usuario.
2. Asignación de usuarios a perfiles
3. Creación y mantenimiento de opciones de menú para perfiles de usuarios

Administración de personal

1. Ingreso y mantenimiento de datos personales
2. Disponibilidad de tiempo de profesores
3. Idiomas por profesor
4. Información salarial de profesores

Administración de Cursos

1. Matriculación de uno o más alumnos para uno o varios cursos
2. Asignación de profesores a cursos
3. Creación de cursos.

4. Ingreso de asistencia de estudiantes.

Administración de calificaciones

1. Creación de aportes
2. Ingreso y mantenimiento de calificaciones
3. Clausura de periodo de evaluación
4. Impresión de cuadro de calificaciones

Reportes

1. Roles de pago de profesores
2. Asistencia de alumnos

2. Especificación del Proyecto (Análisis y Diseño)

2.1 Análisis de Requerimientos

El propósito del proyecto es desarrollar un proyecto de tesis de grado que consiste de un software que hace uso de las nuevas tecnologías AJAX y otras tecnologías Open Source siguiendo la metodología RUP y así permita automatizar los procesos más críticos de una institución como el Lincoln College of Foreign Languages

Diagramas de Casos de Uso

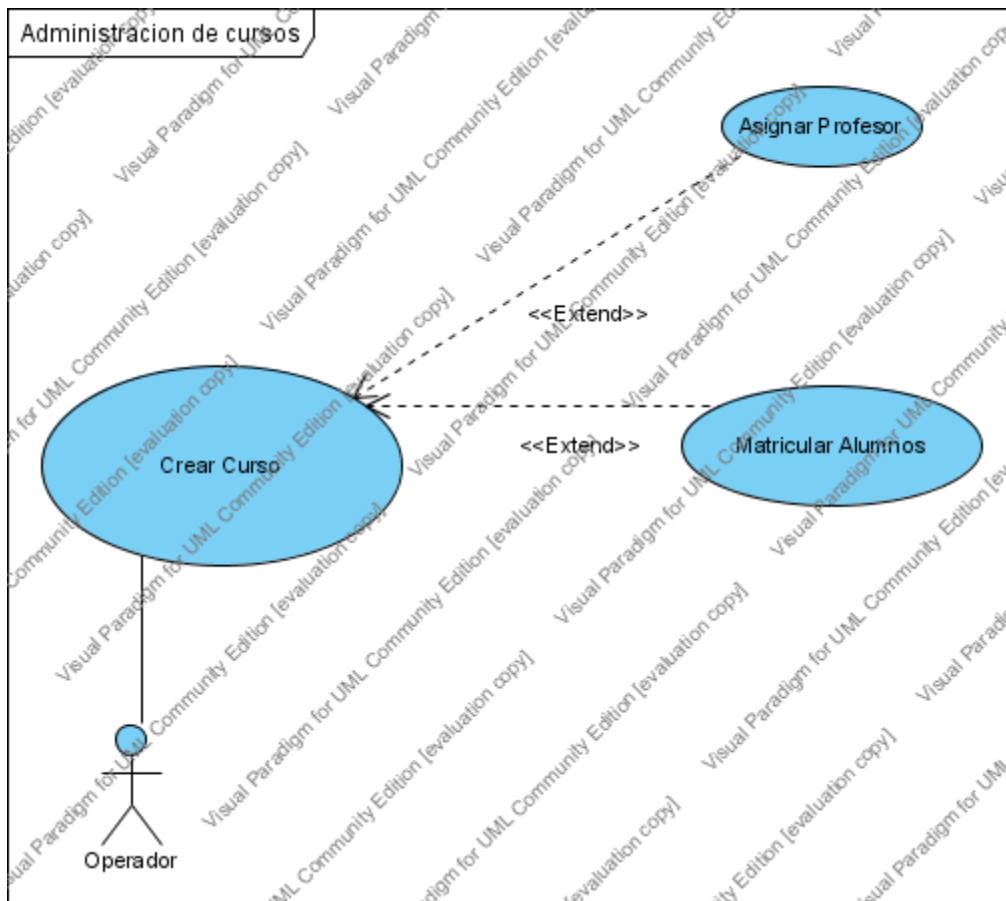
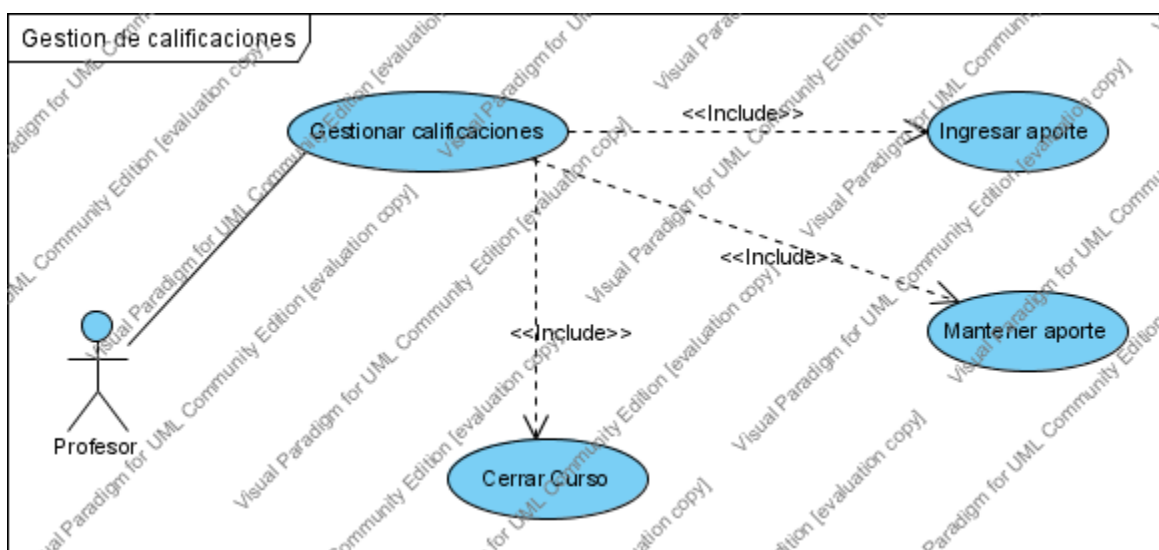


Gráfico 3.5: Casos de Uso: Administración de cursos



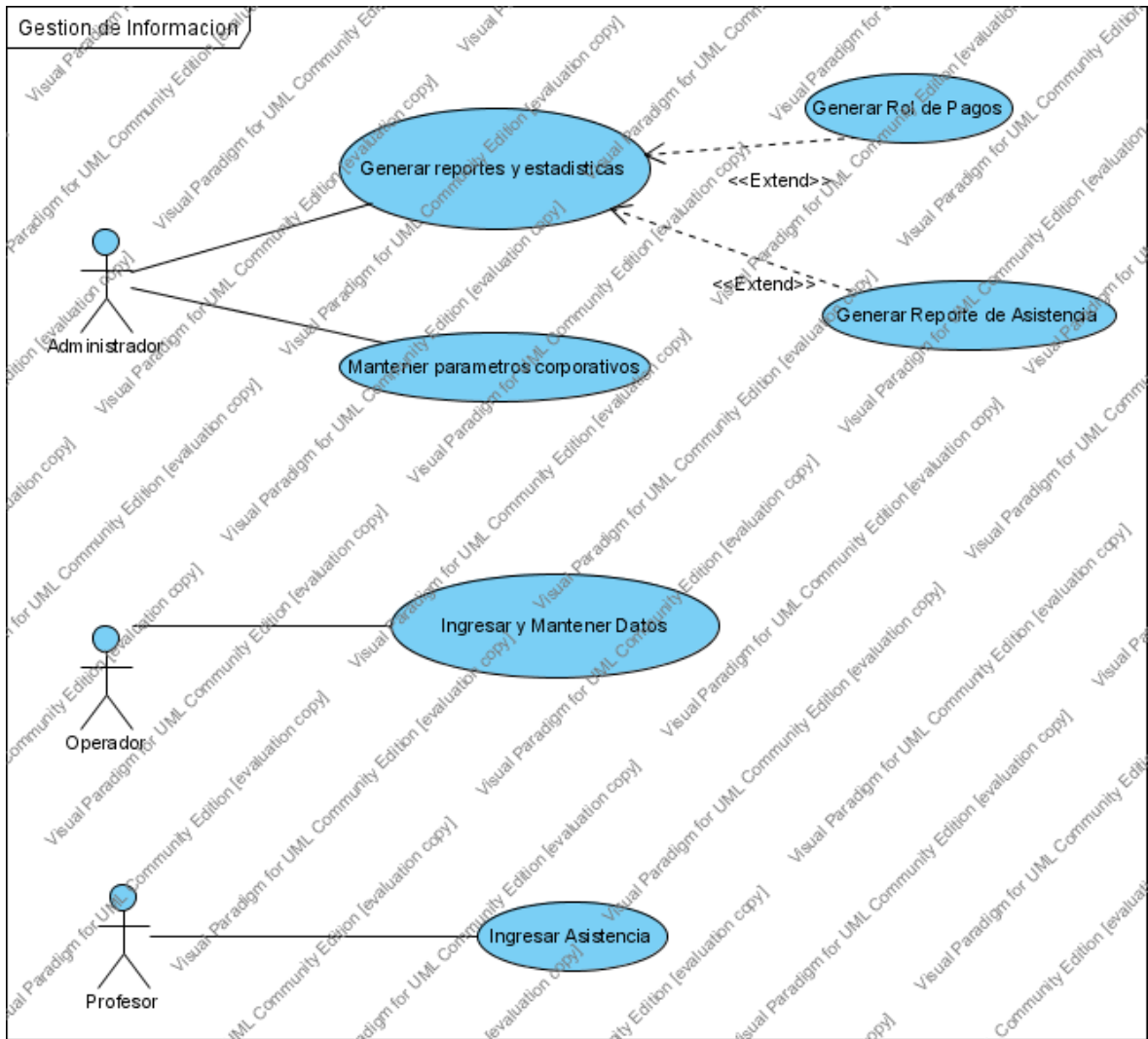
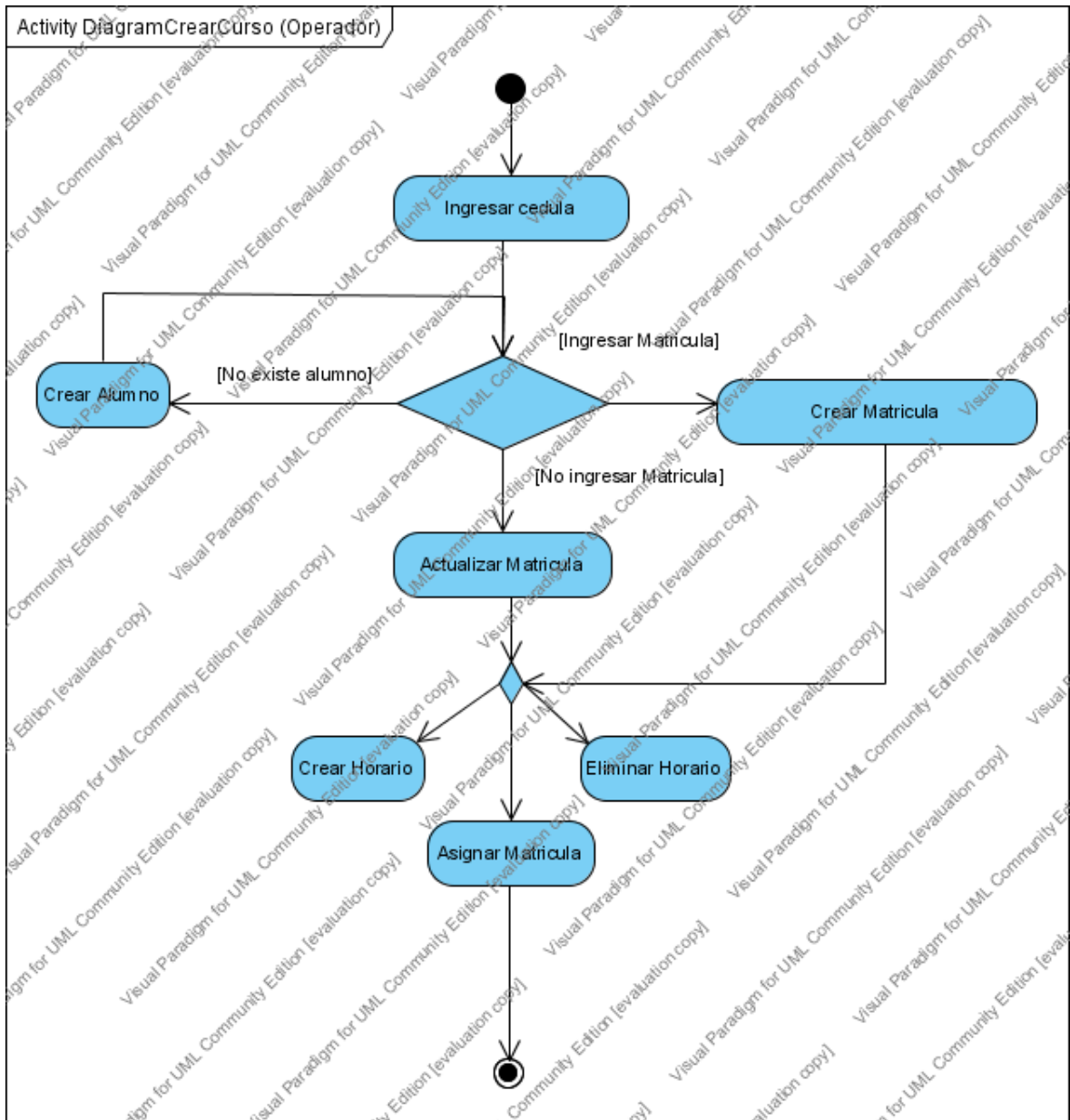
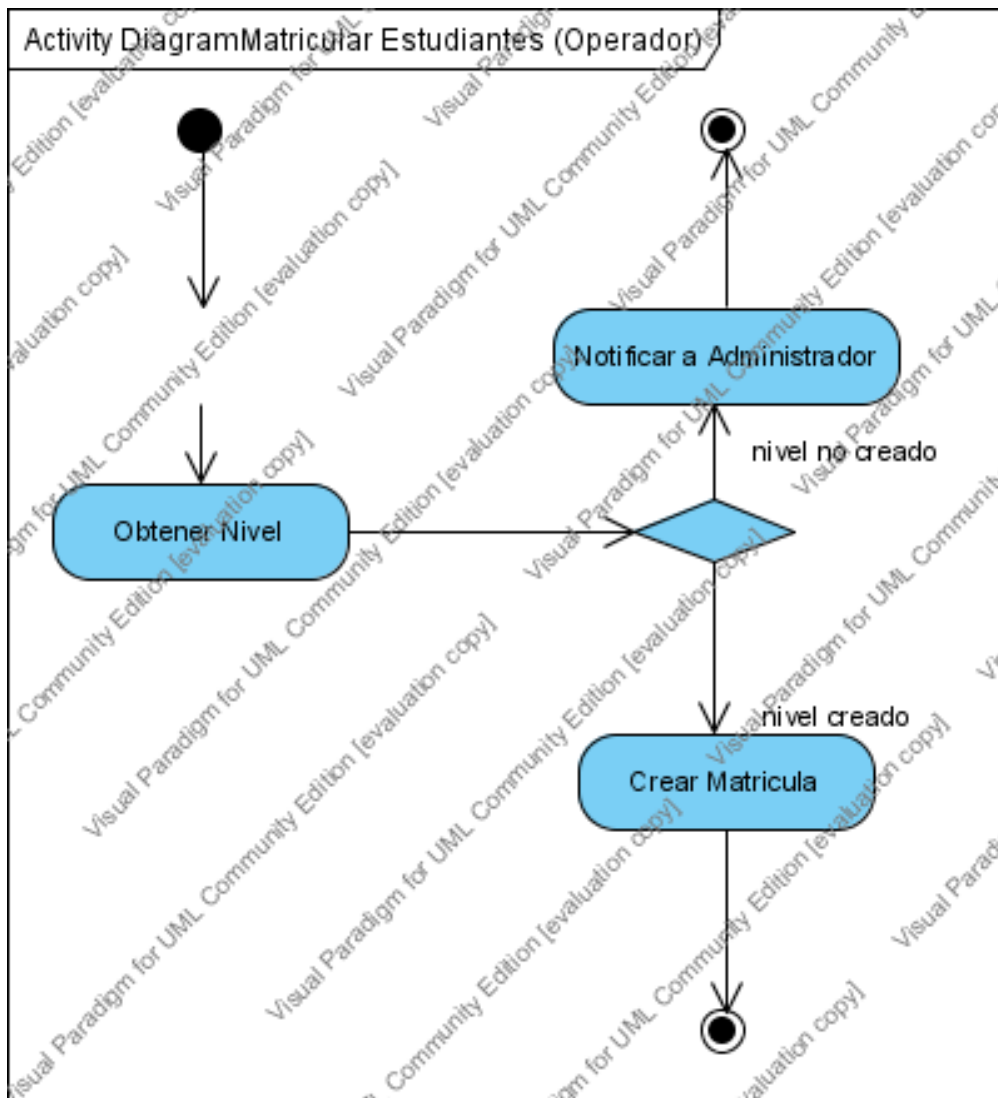
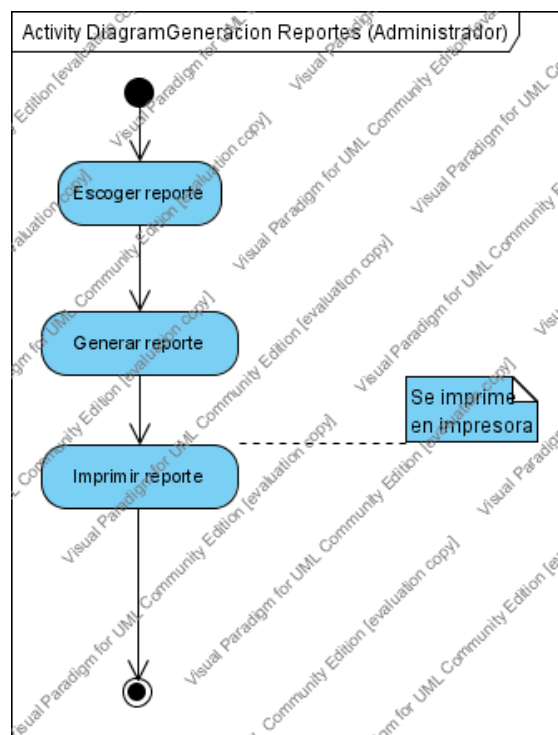
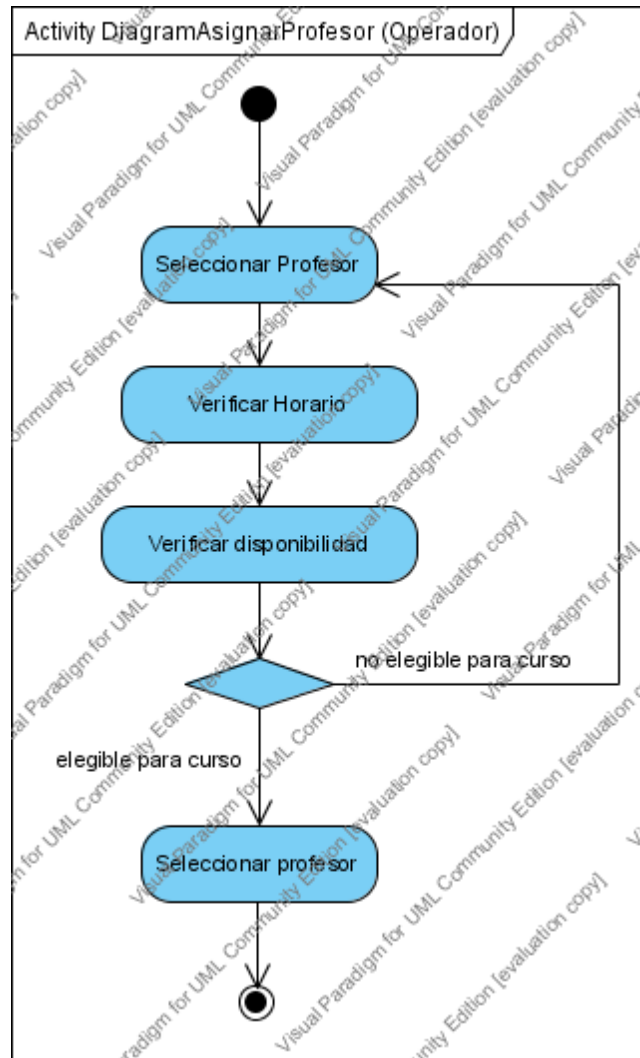
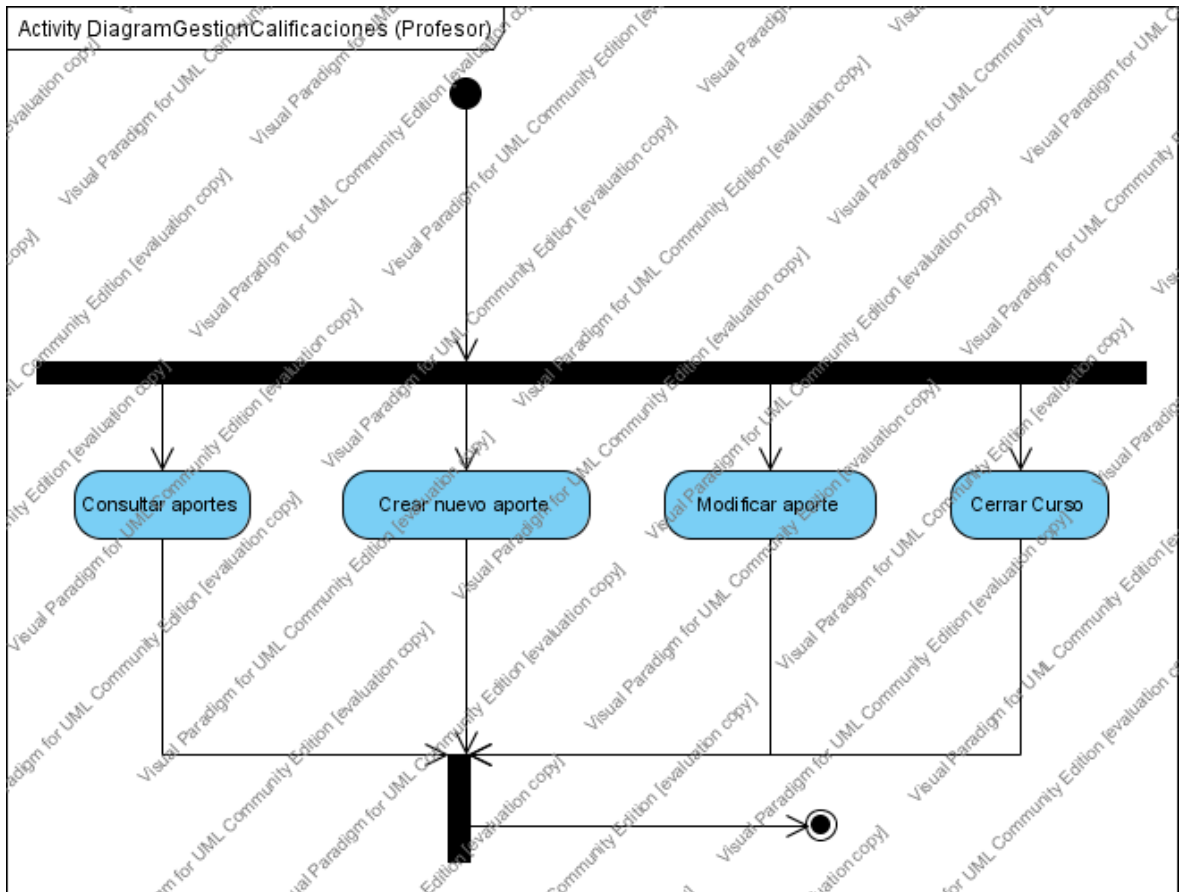


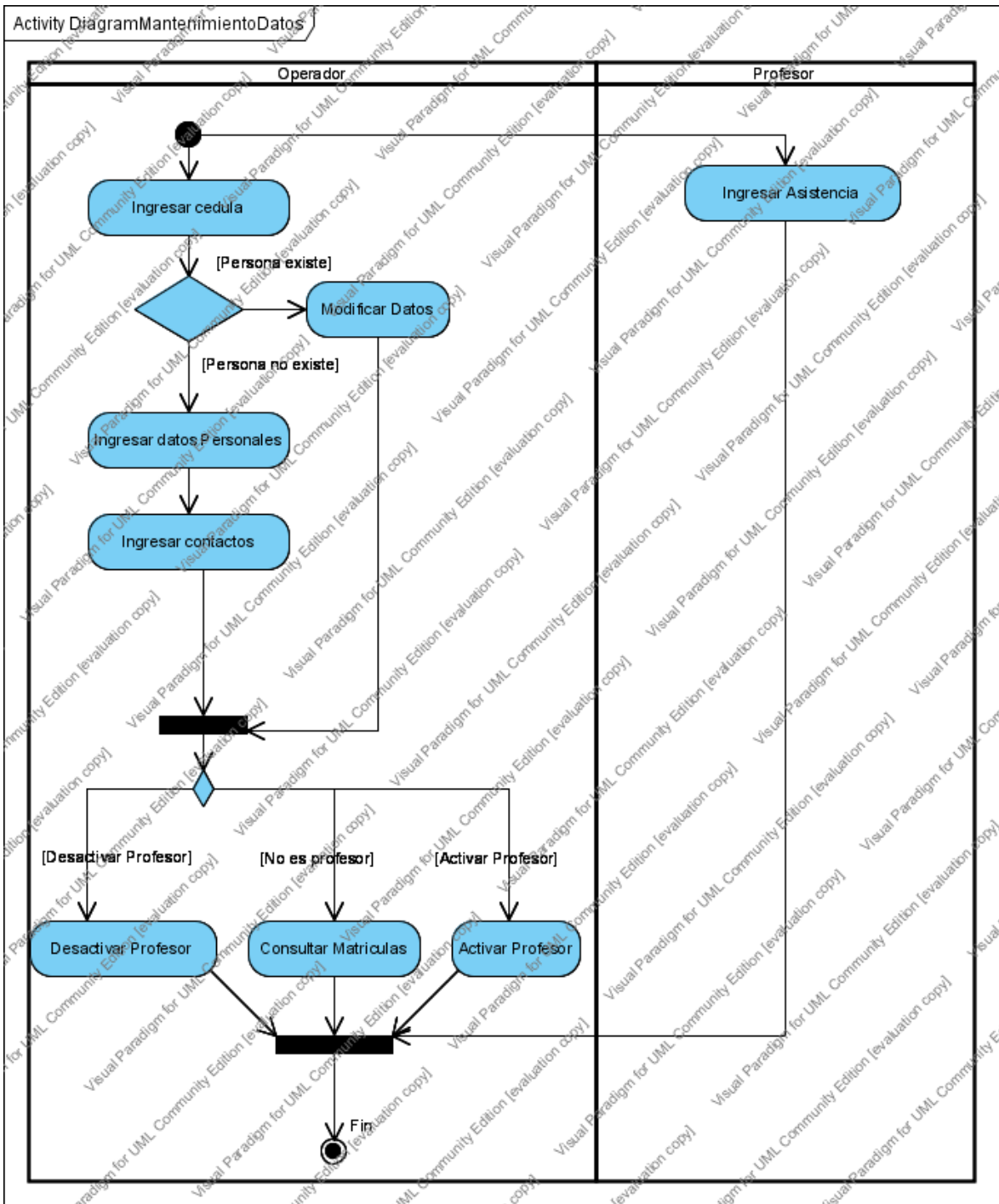
Diagrama de Actividades de los Casos de Uso

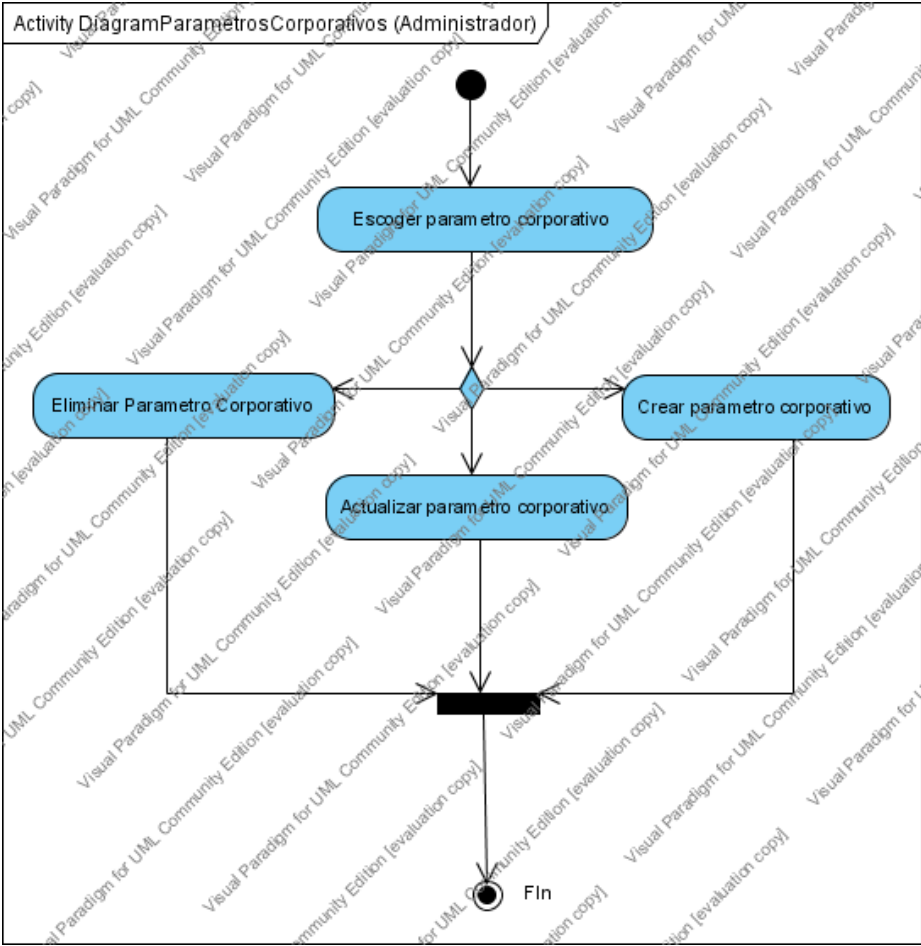
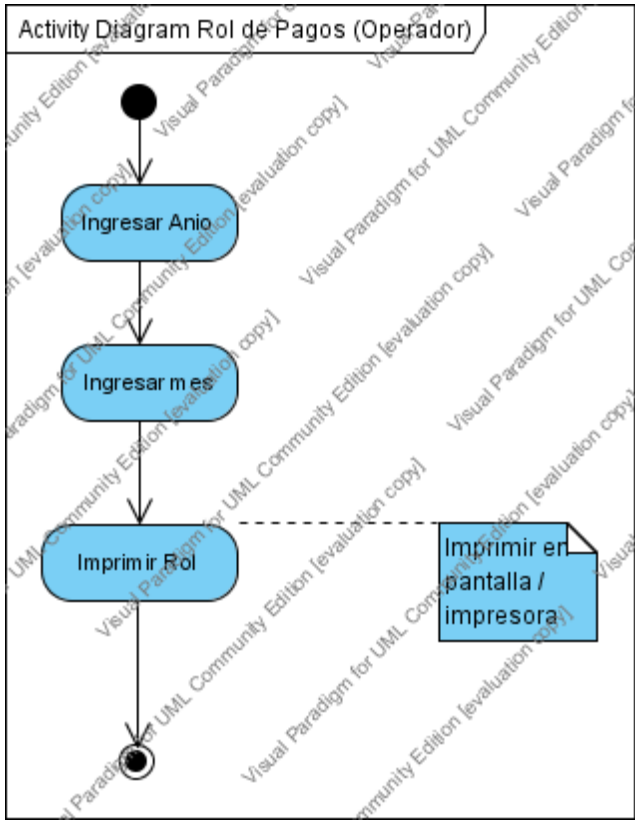


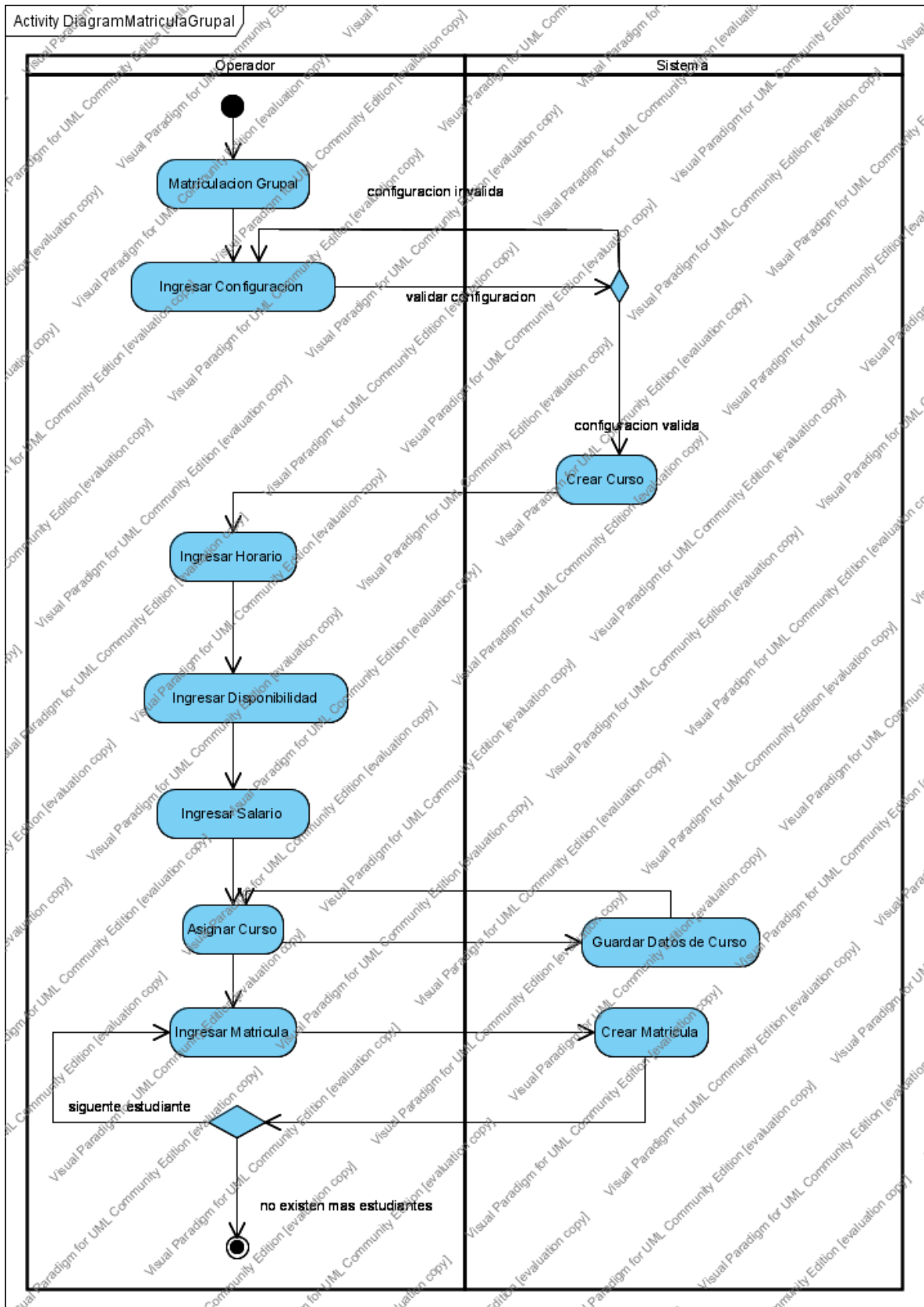












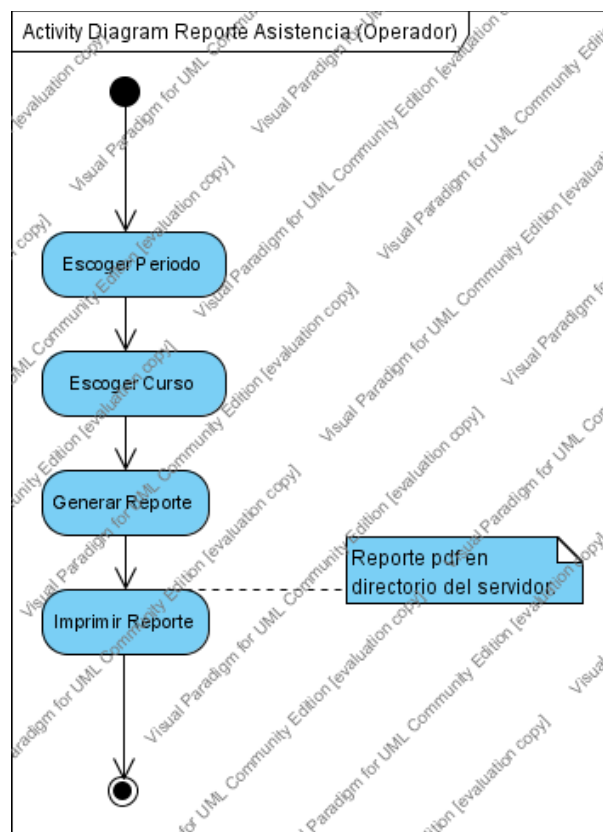
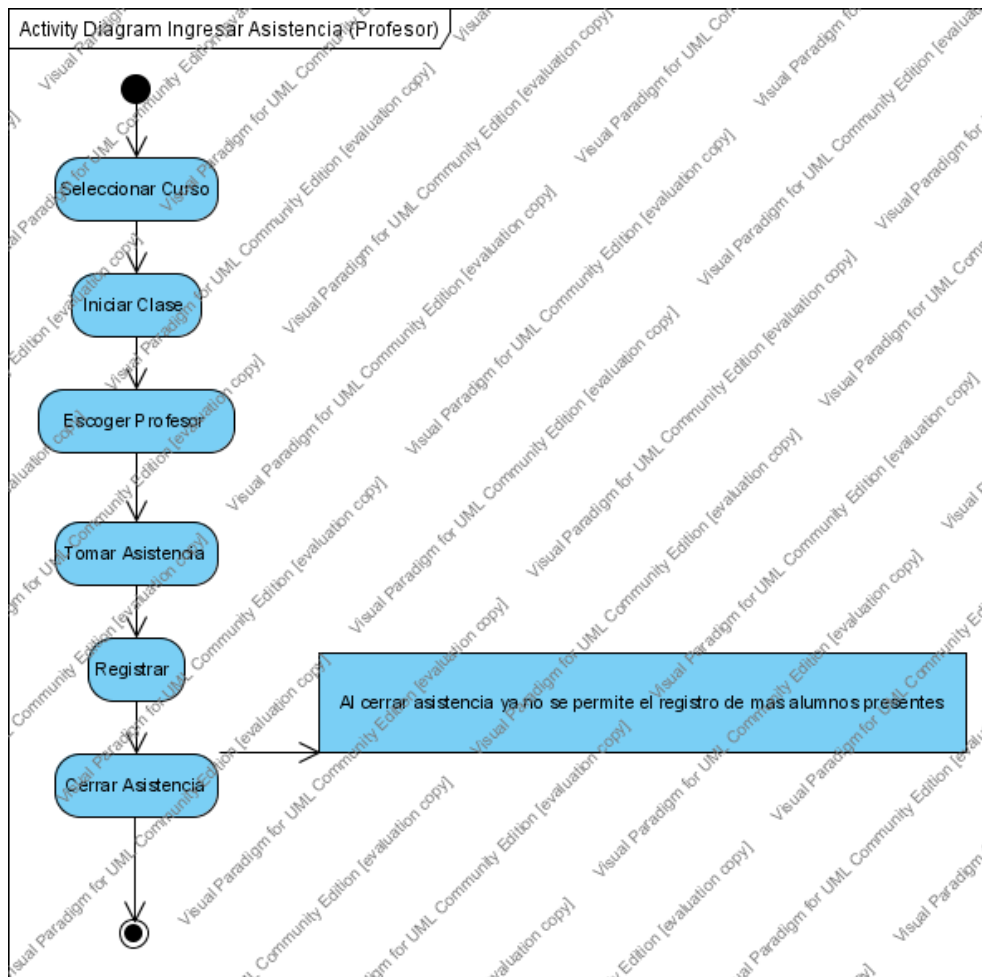
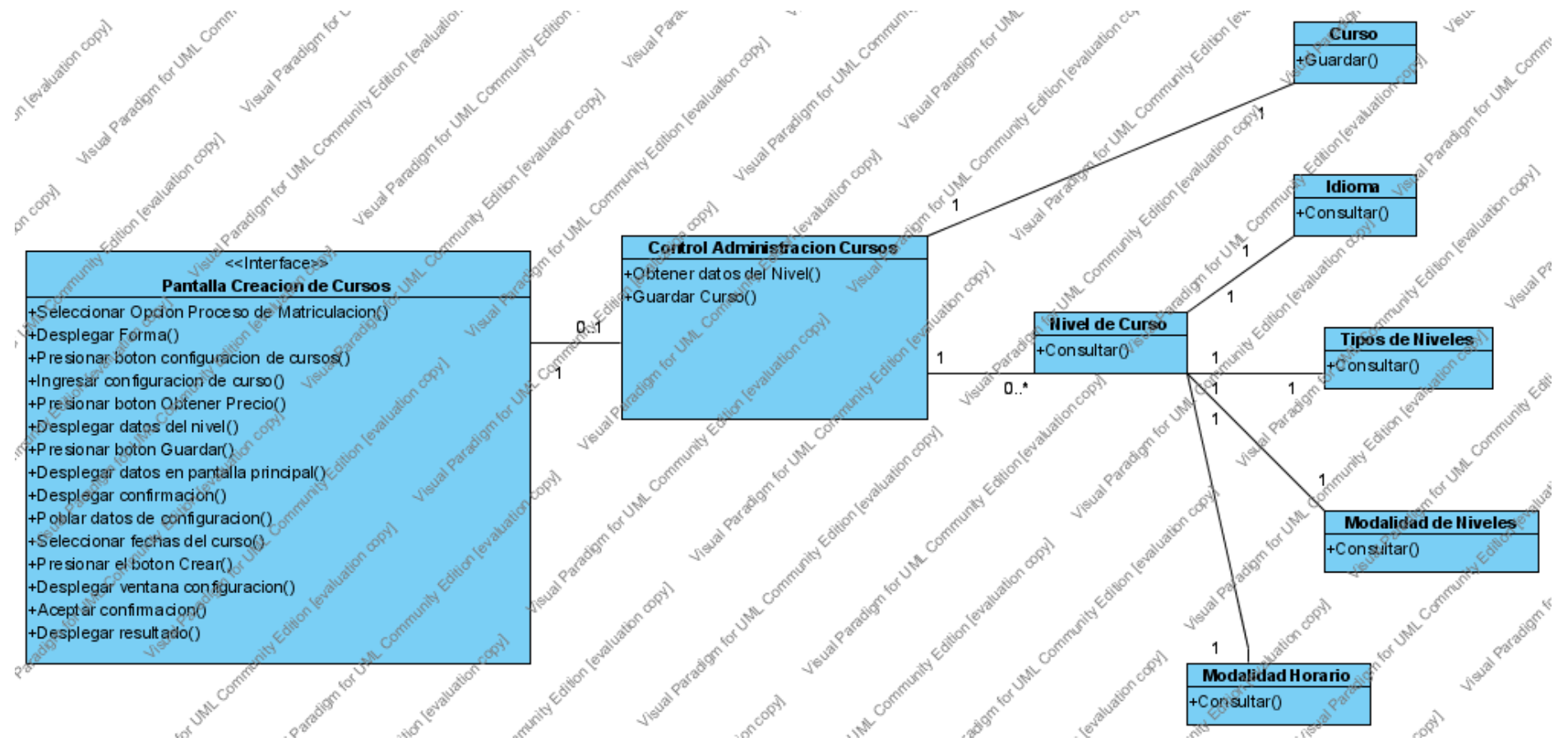
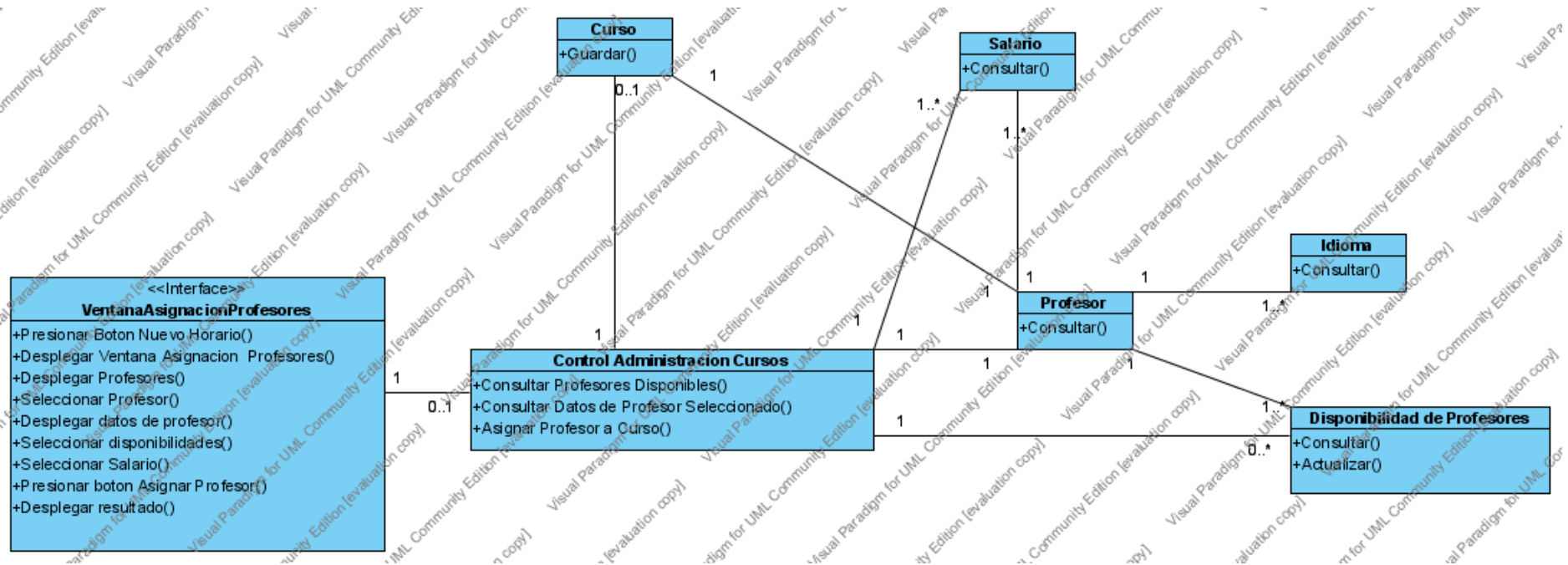


Diagrama de Clases de Análisis

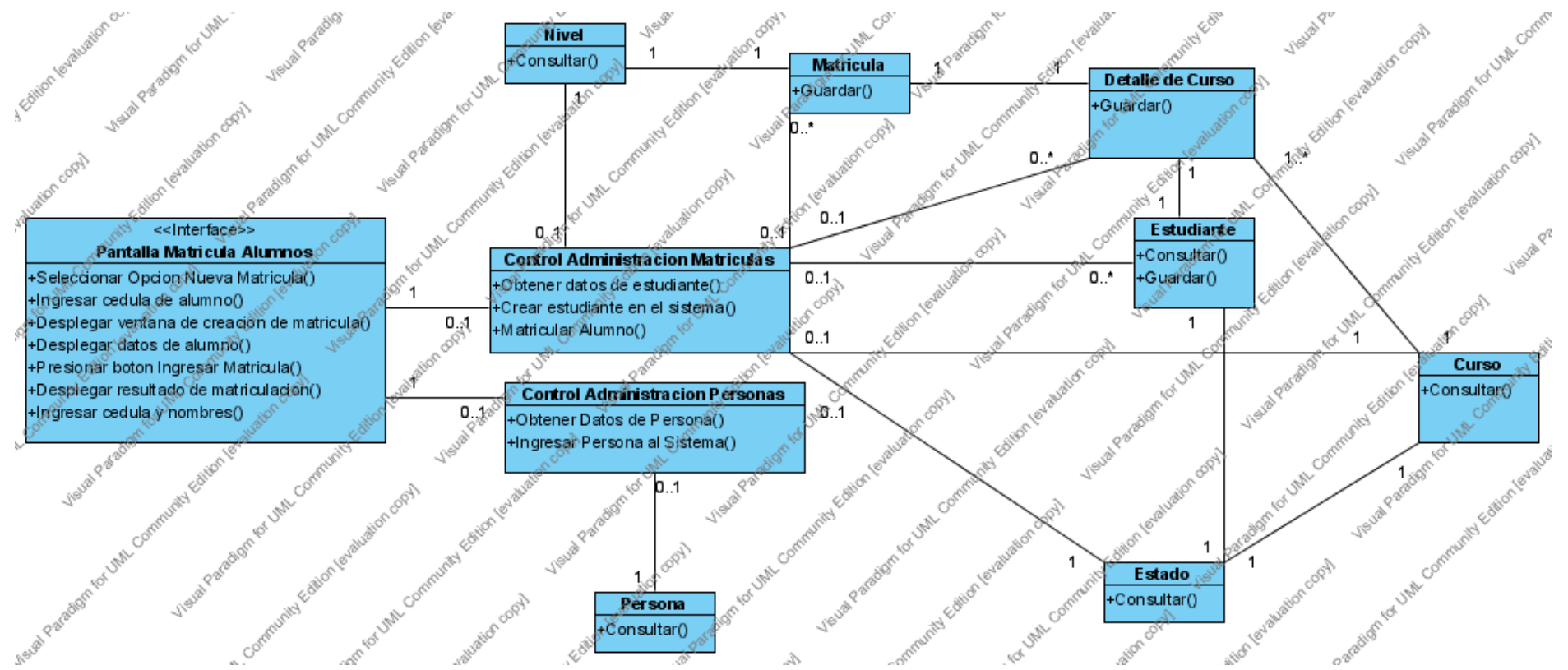
Crear Curso



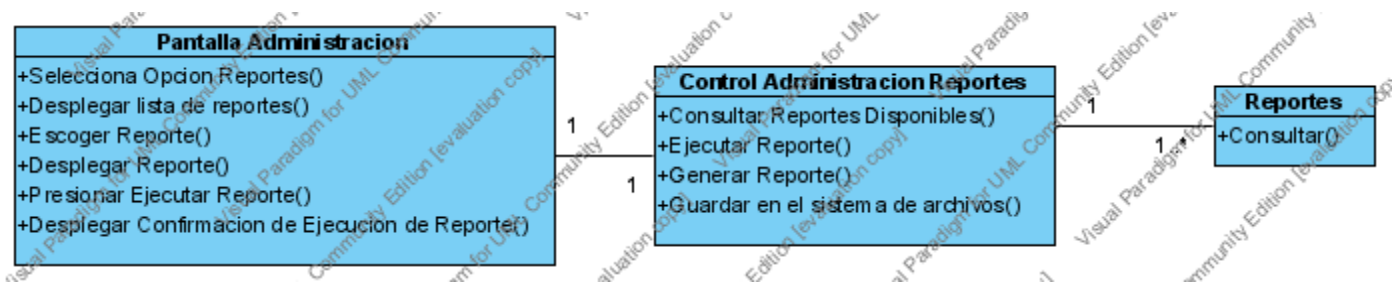
Asignar Profesor



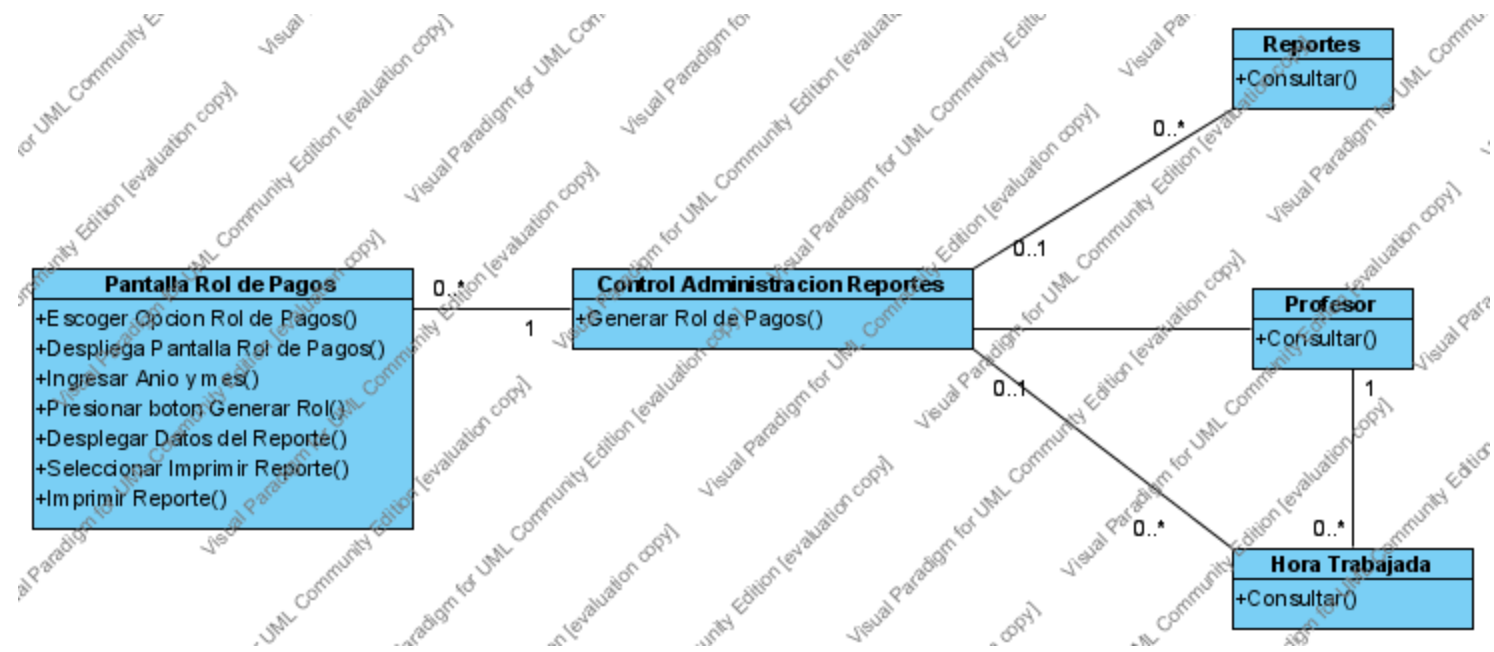
Matricular Alumnos



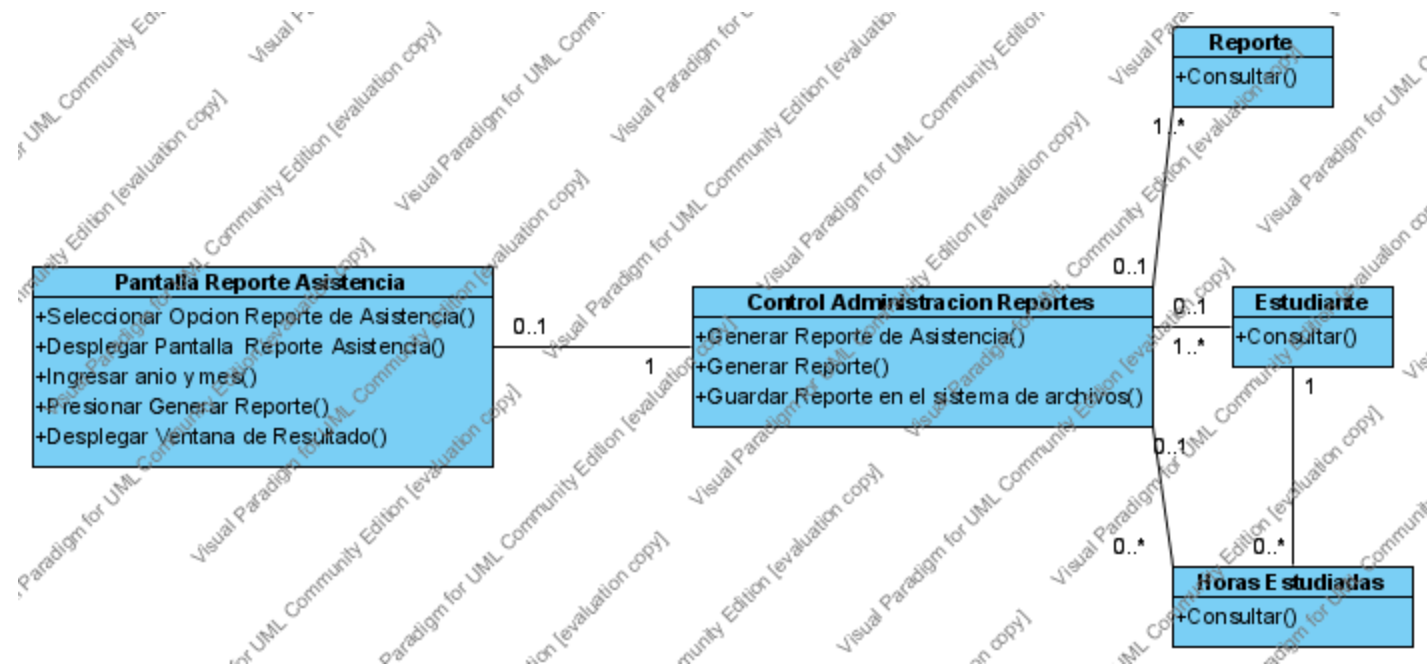
Generar Reportes y Estadísticas



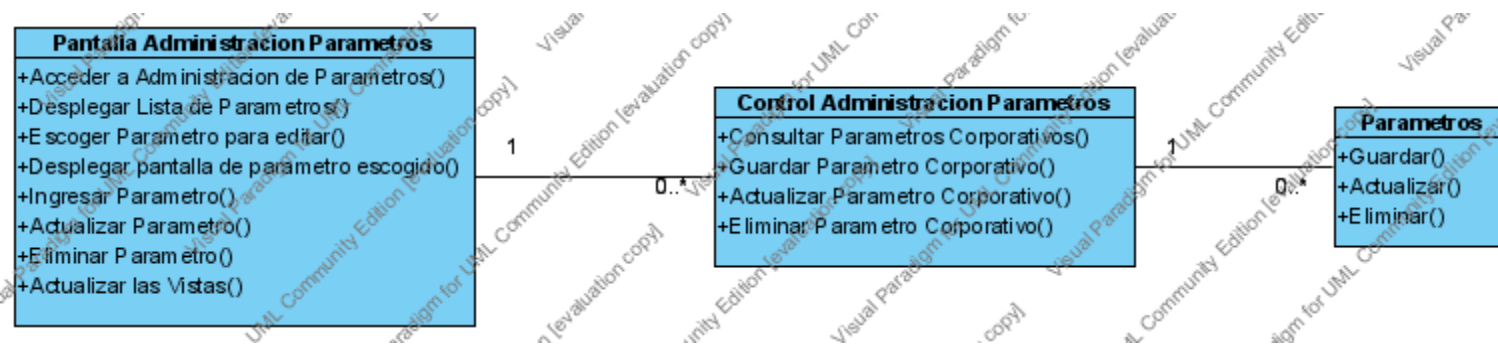
Generar Rol de Pagos



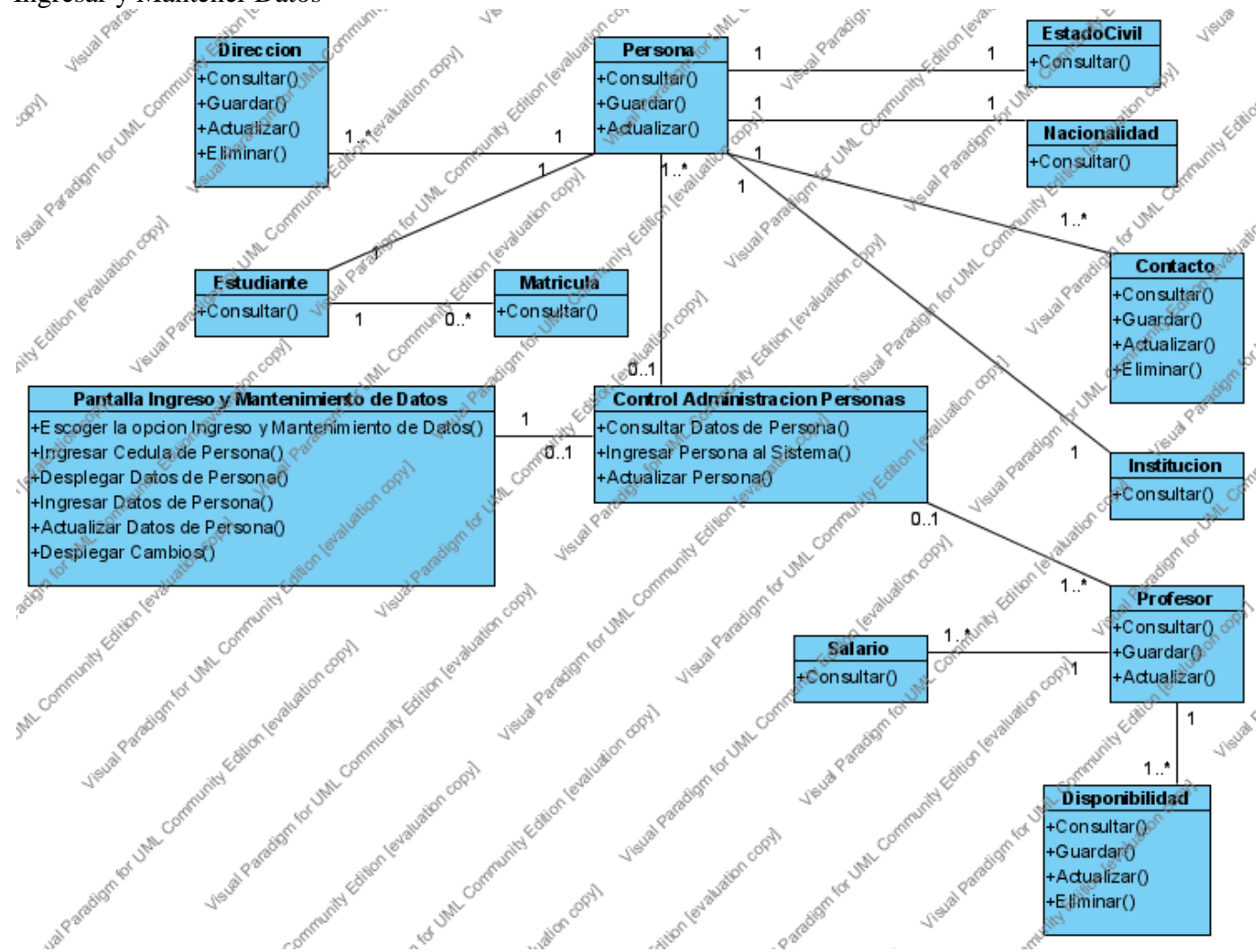
Generar Reporte de Asistencia



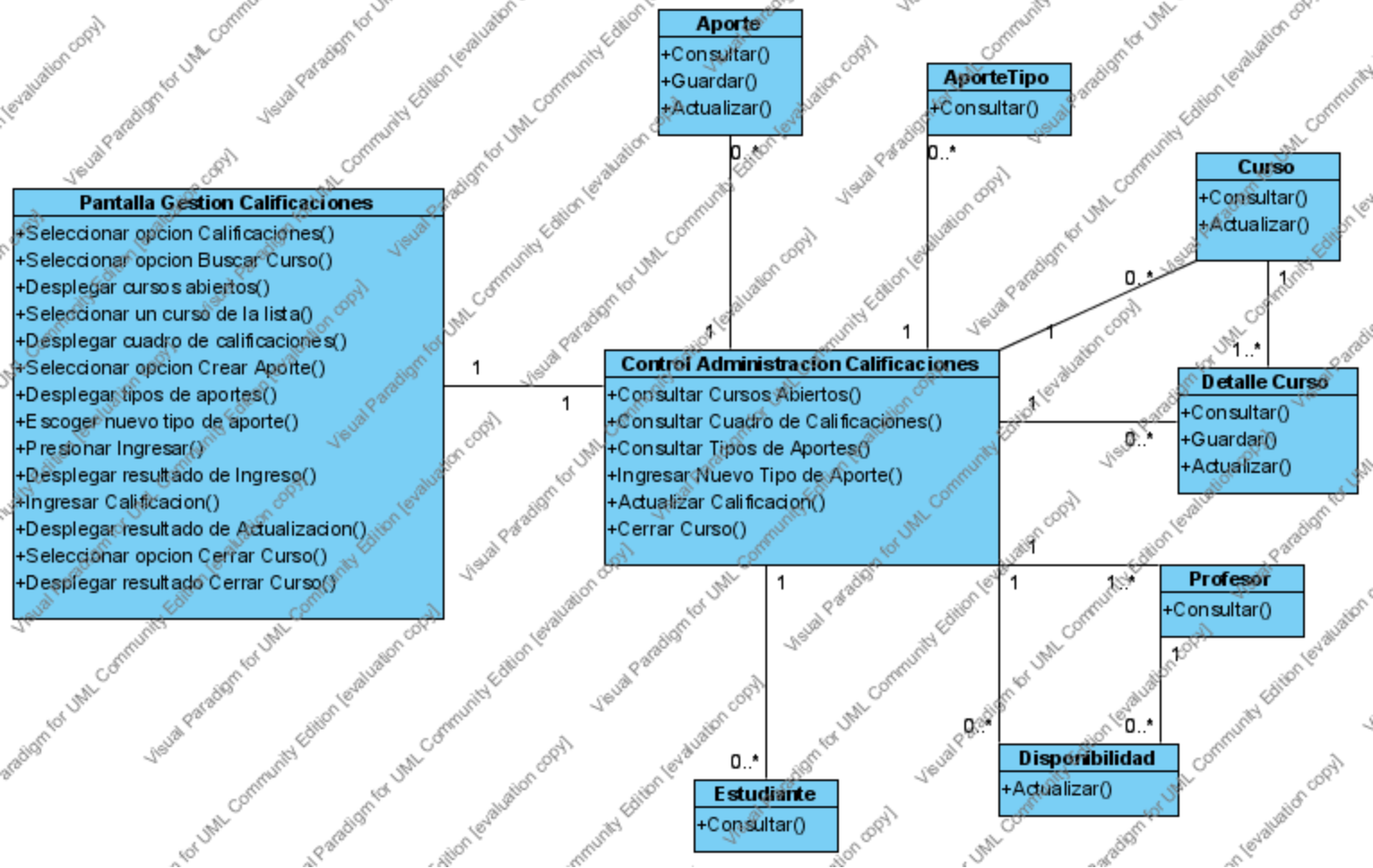
Mantener Parámetros Corporativos



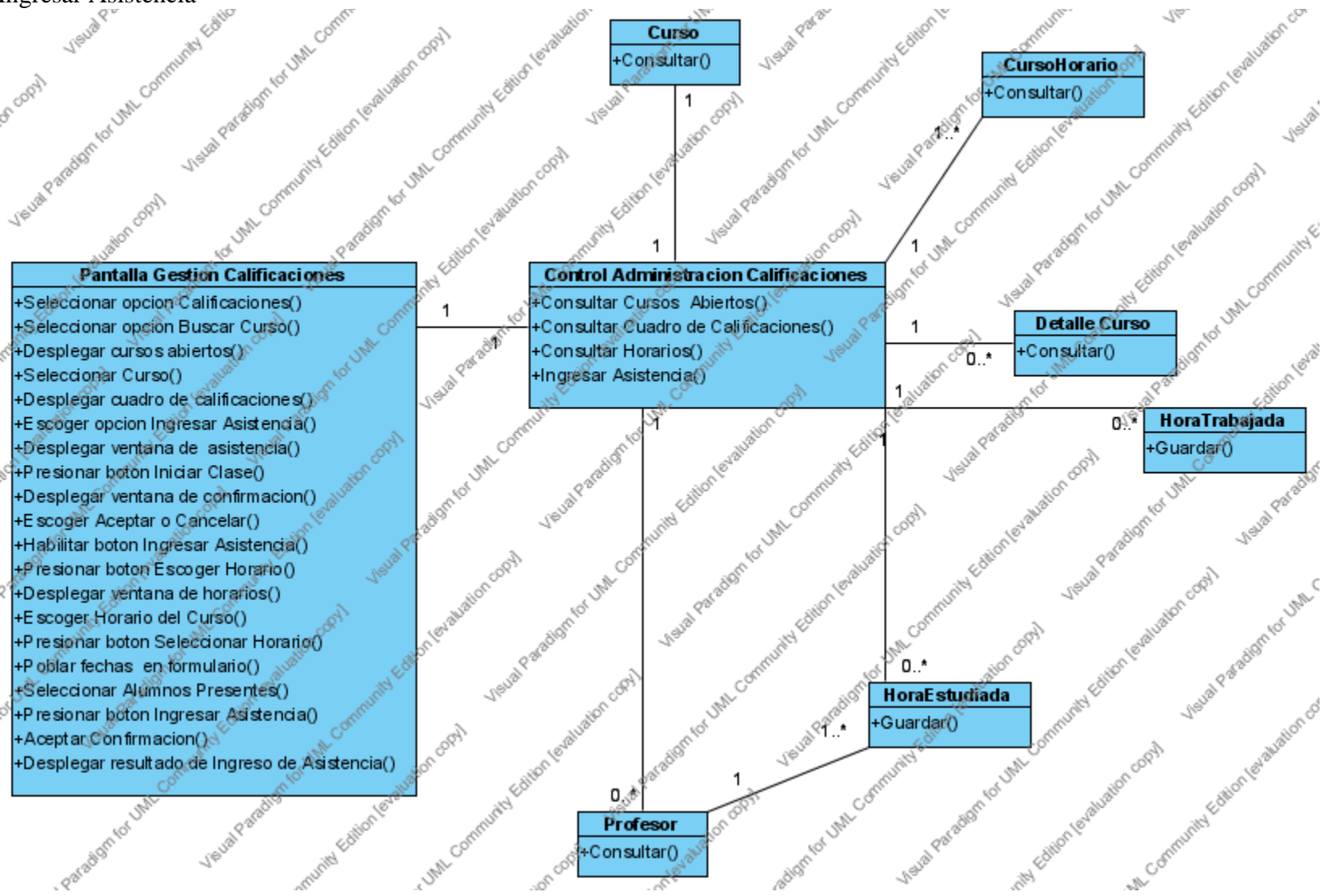
Ingresar y Mantener Datos



Gestionar Calificaciones



Ingresar Asistencia



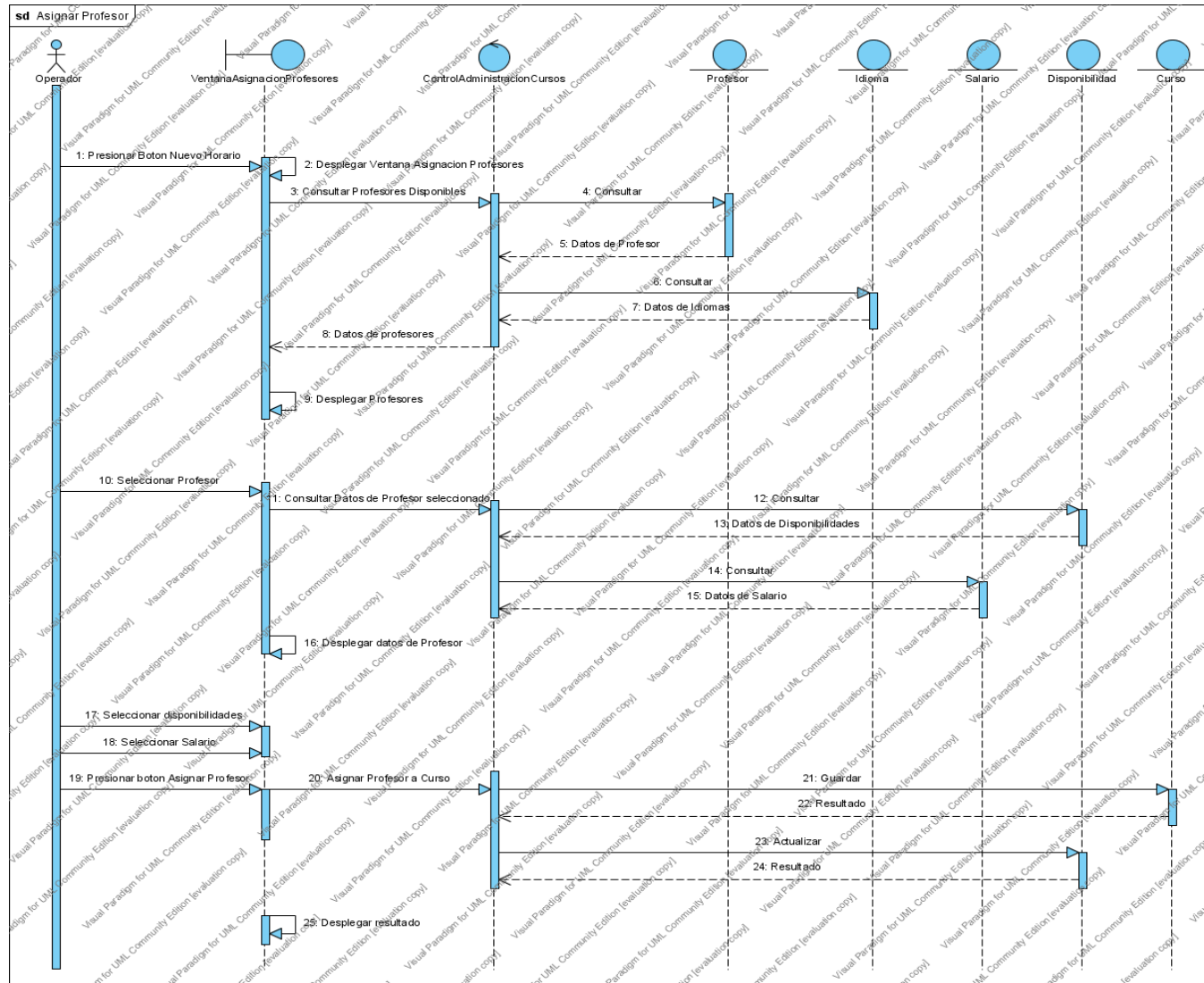


Gráfico 3.7: Diagrama de Secuencia: Asignar Profesor

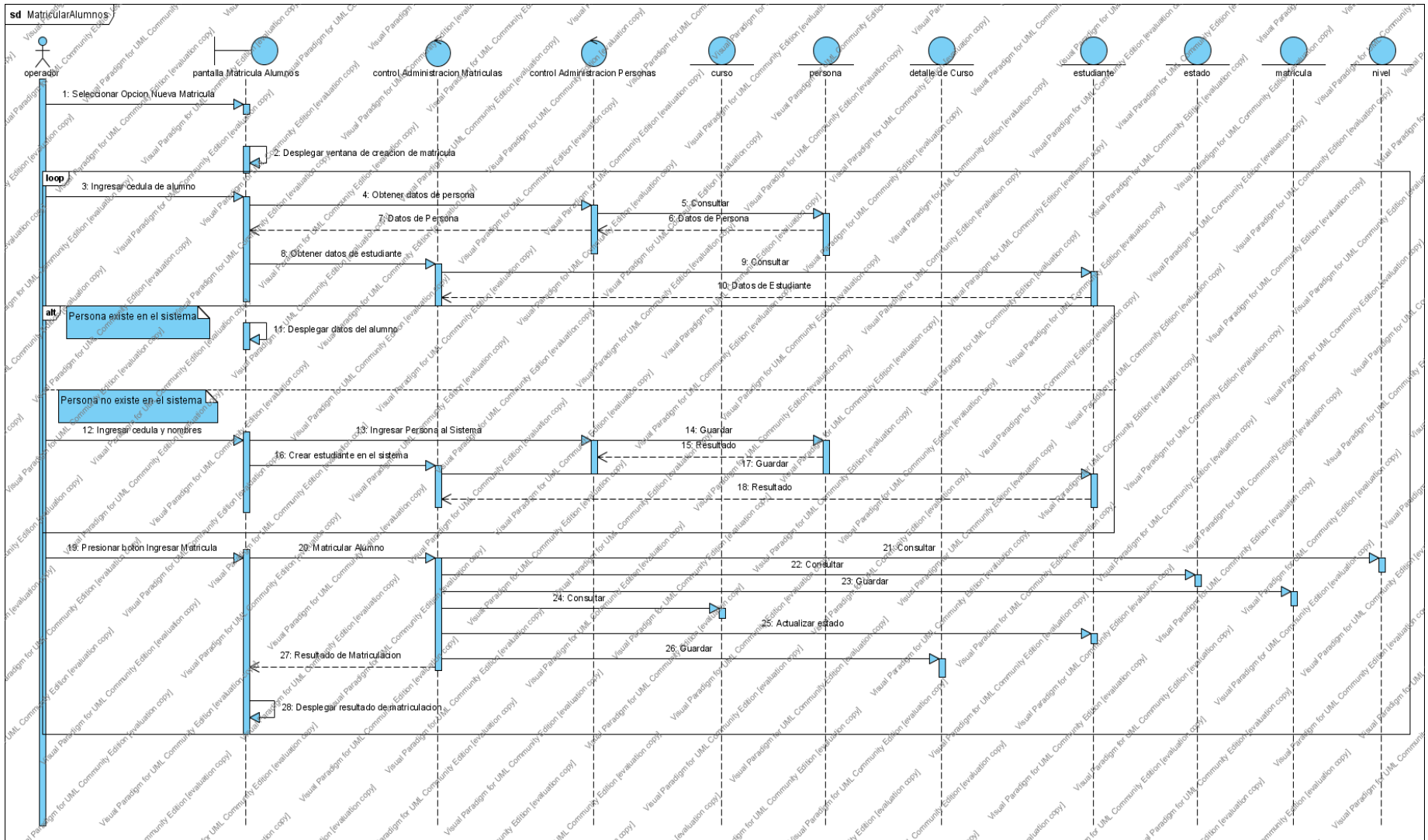


Gráfico 3.8: Diagrama de Secuencia: Matricular Alumnos

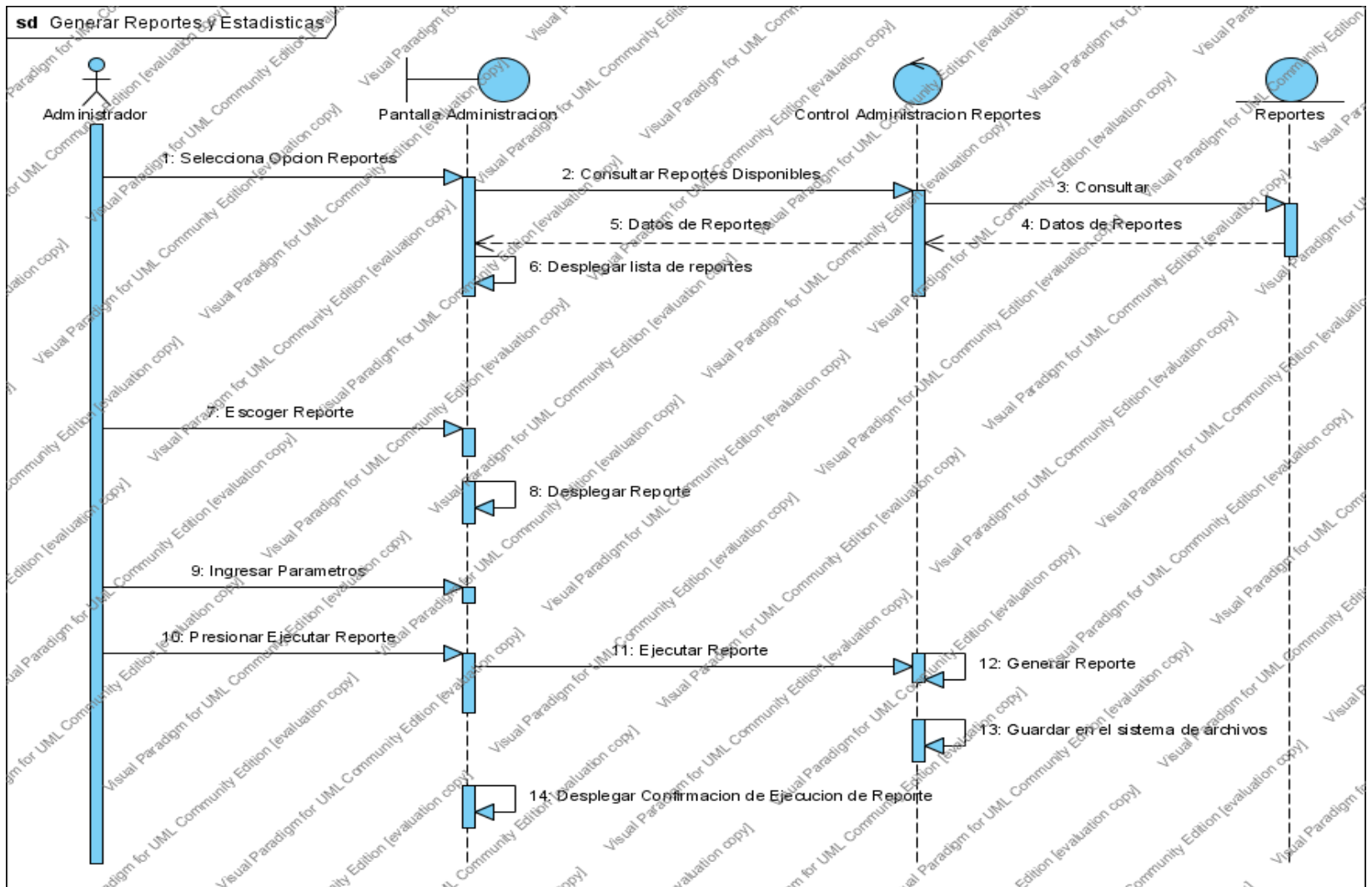


Gráfico 3.9 Diagrama de Secuencia: Generar Reportes y Estadísticas

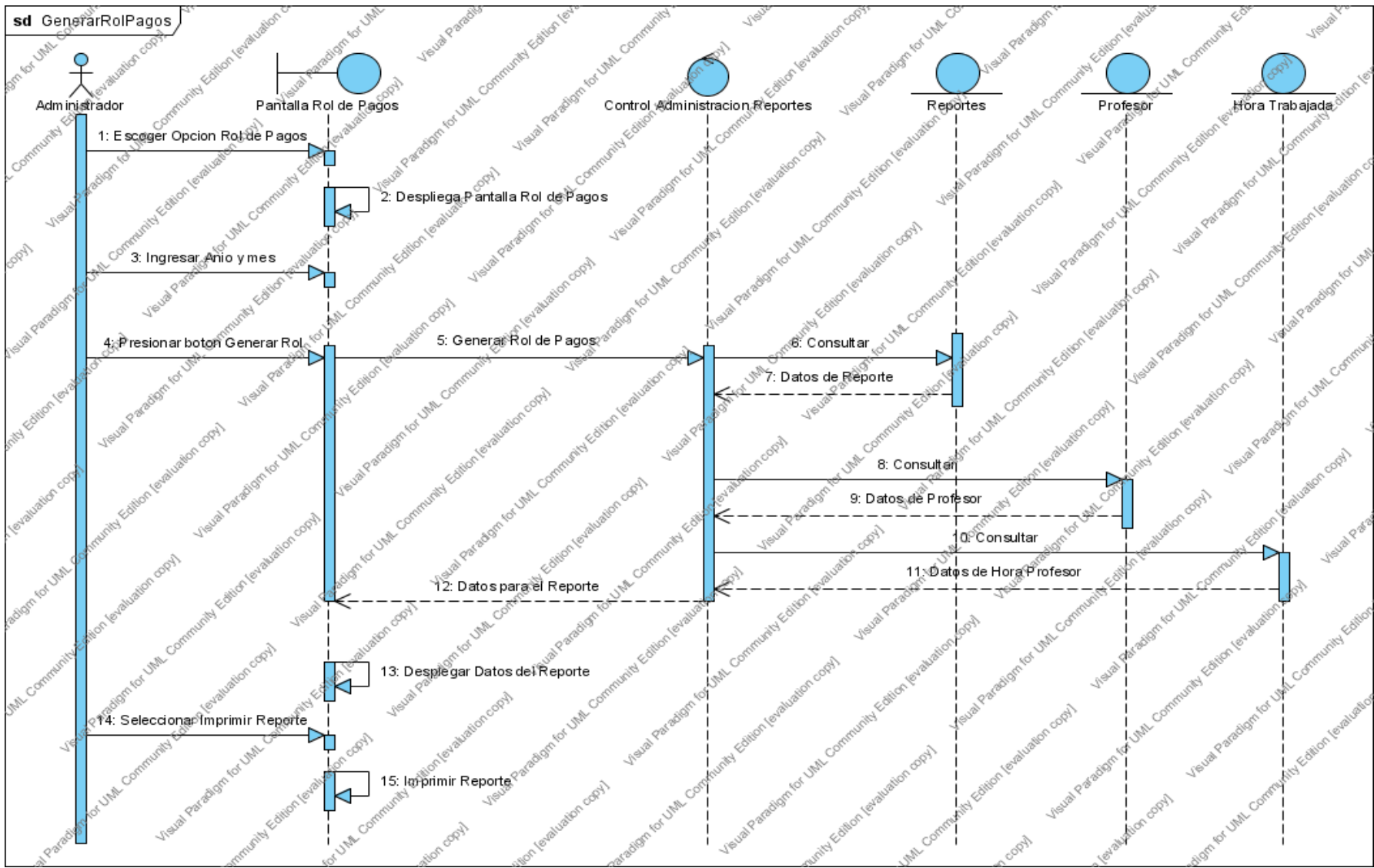


Gráfico 3.10: Generar Rol de Pagos

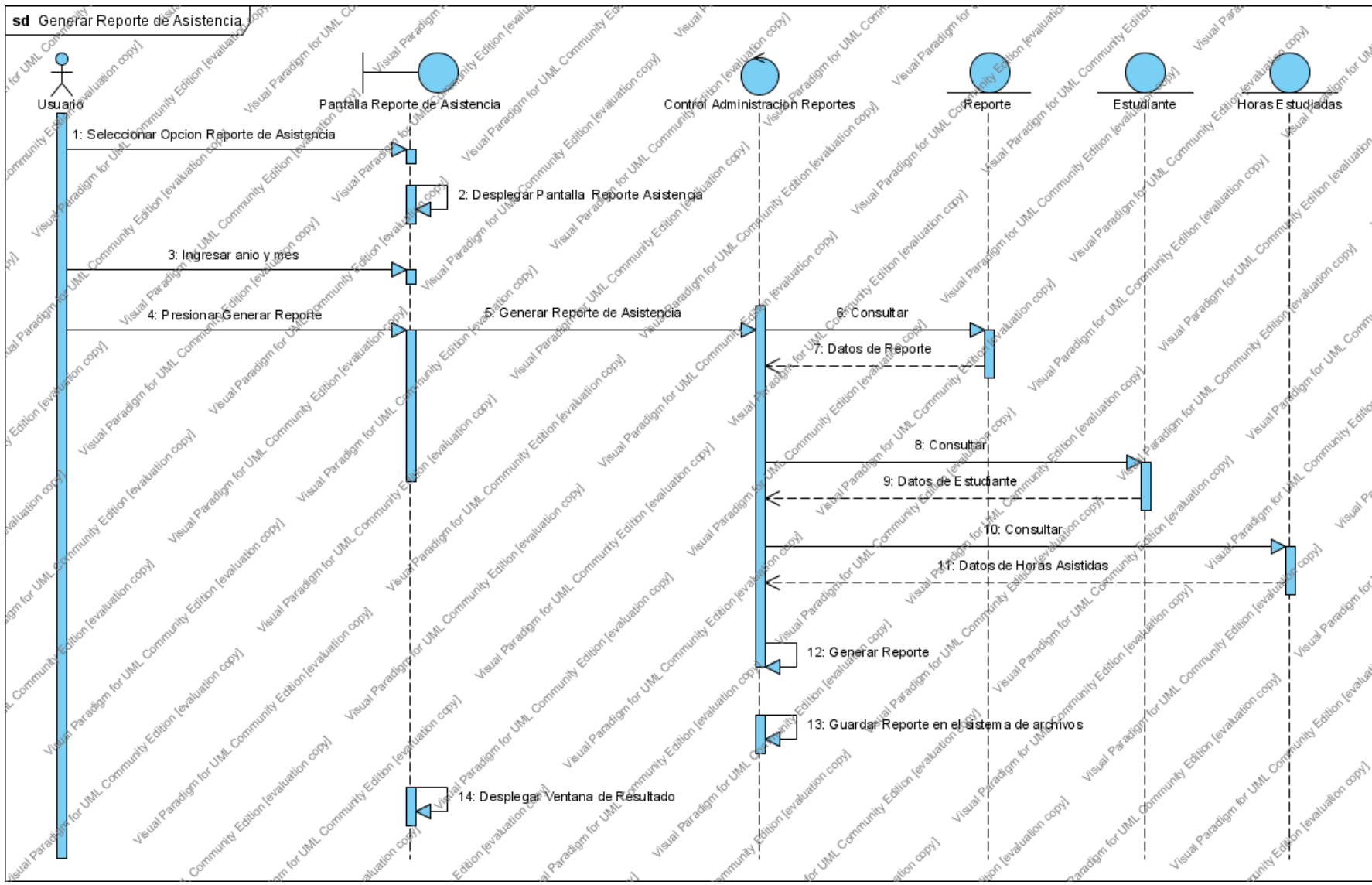


Gráfico 3.11: Generar Reporte de Asistencia

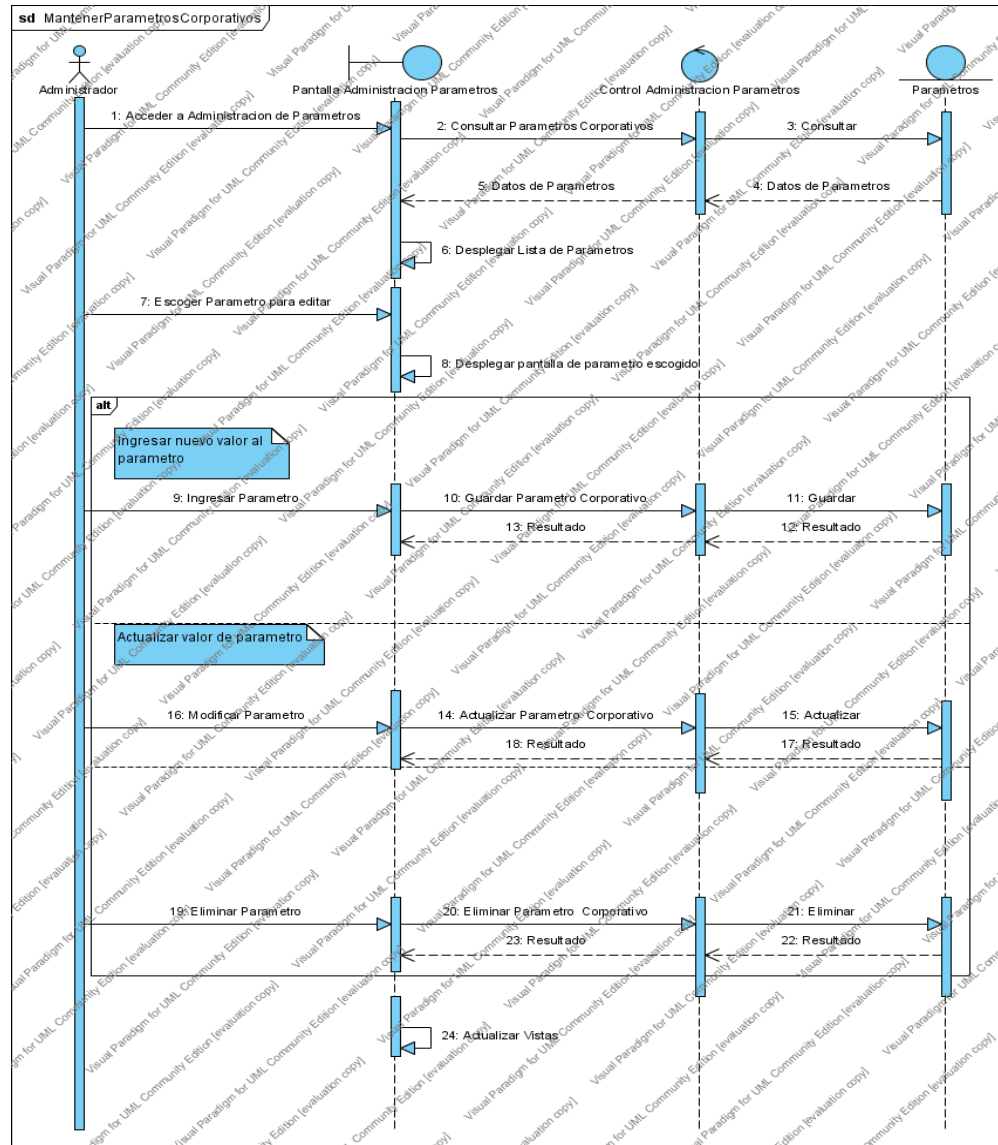


Gráfico 3.12: Mantener Parámetros Corporativos

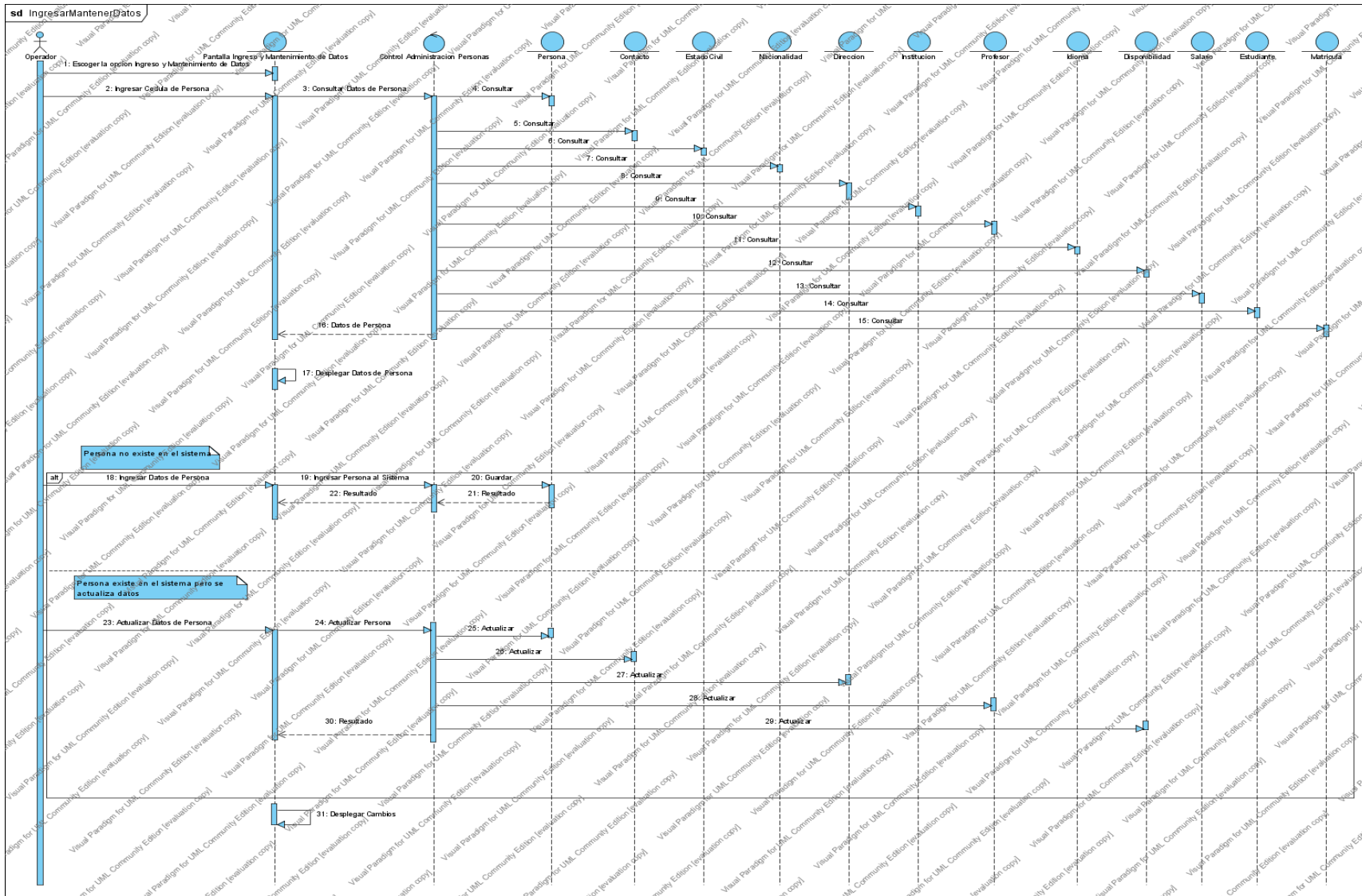


Gráfico 3.13: Ingresar y Mantener Datos

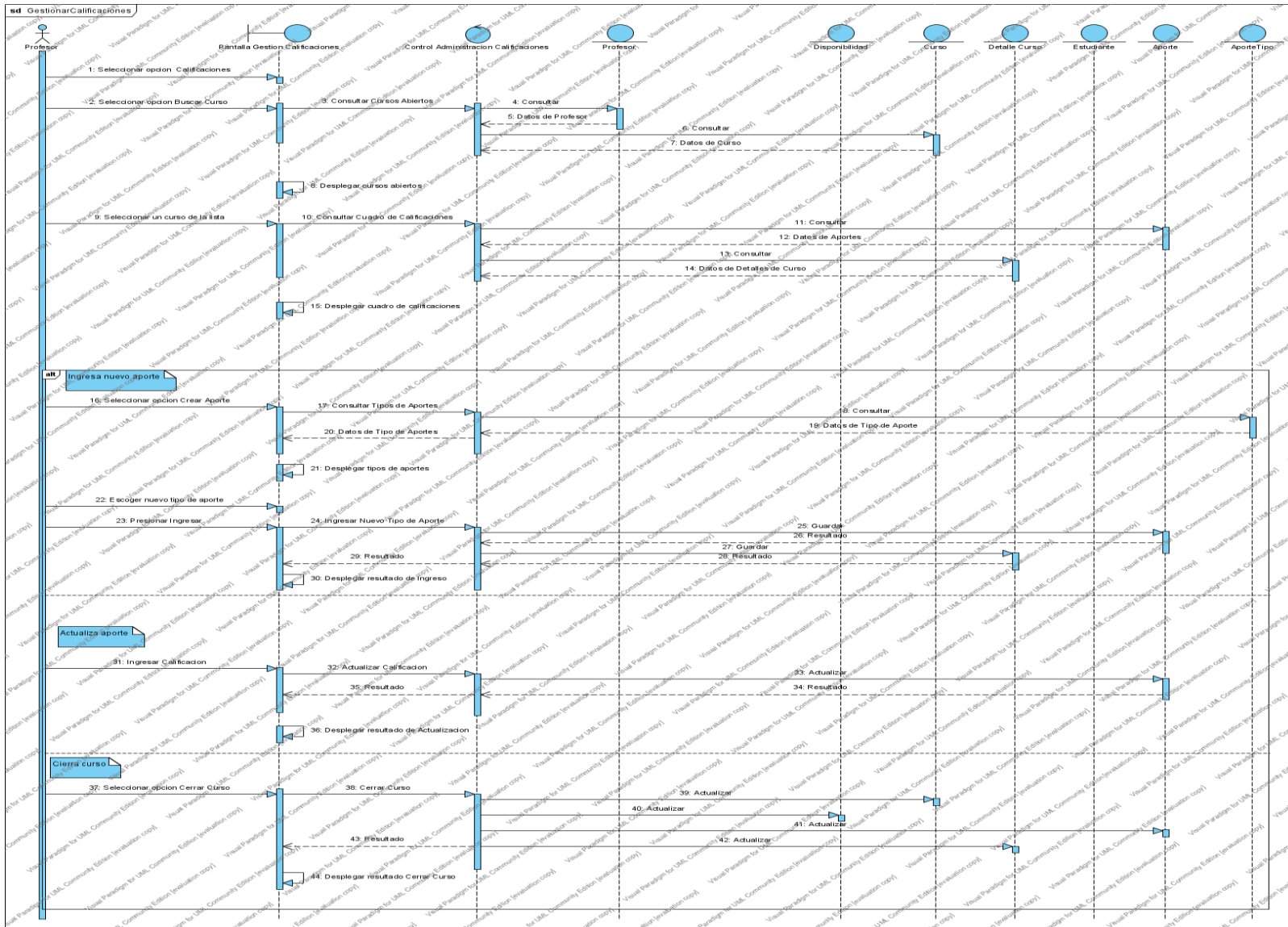


Gráfico 3.14: Gestionar Calificaciones

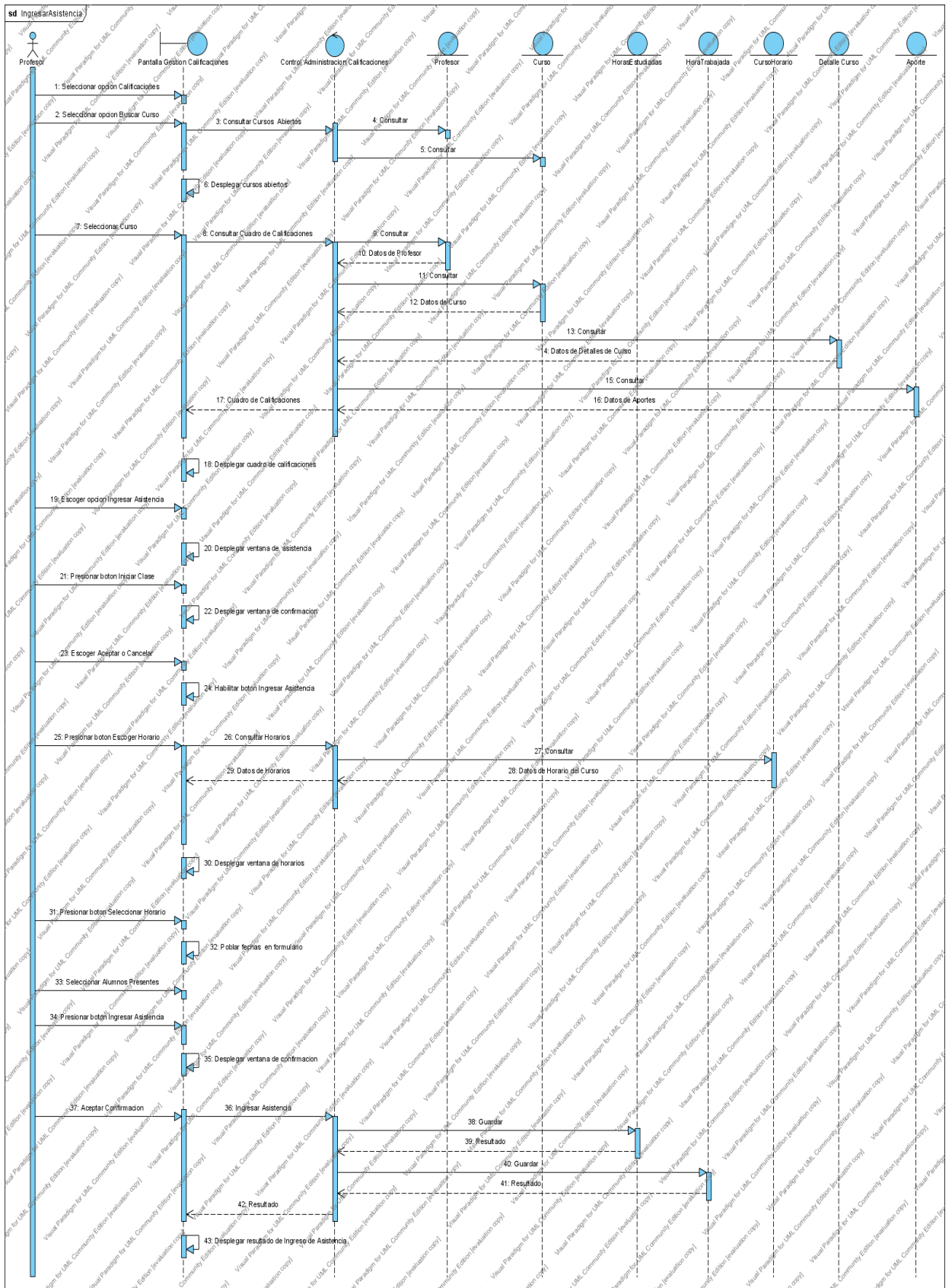
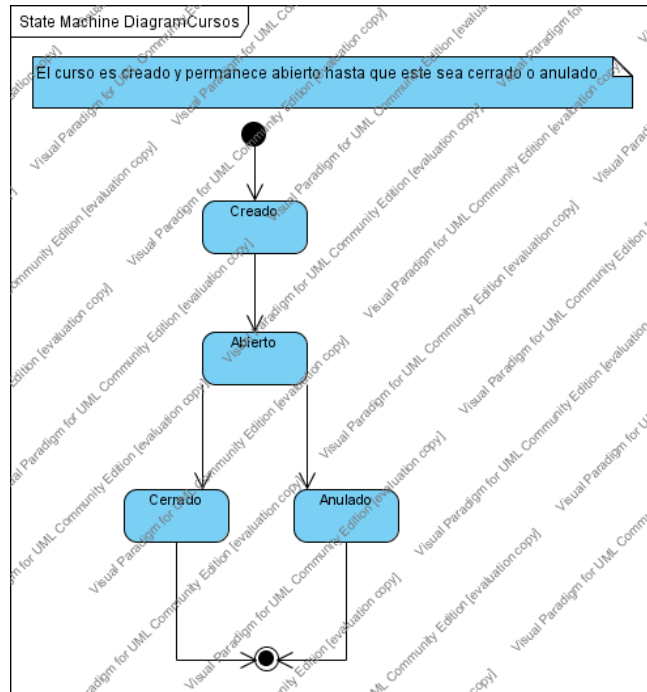


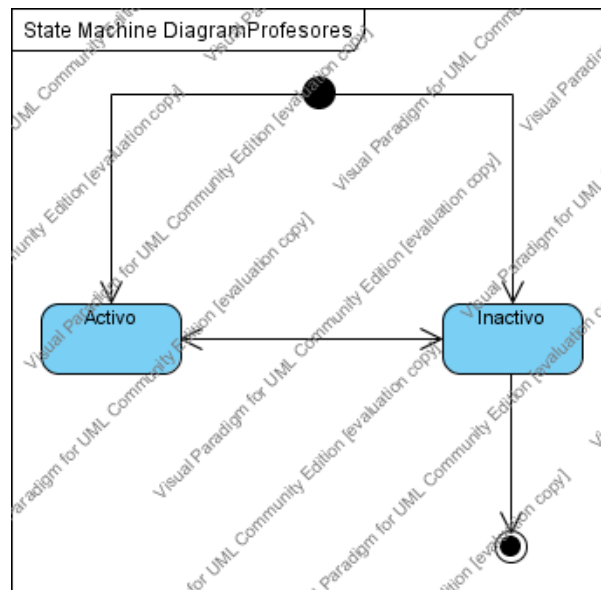
Gráfico 3.15: Ingresar Asistencia

Diagrama de Estados

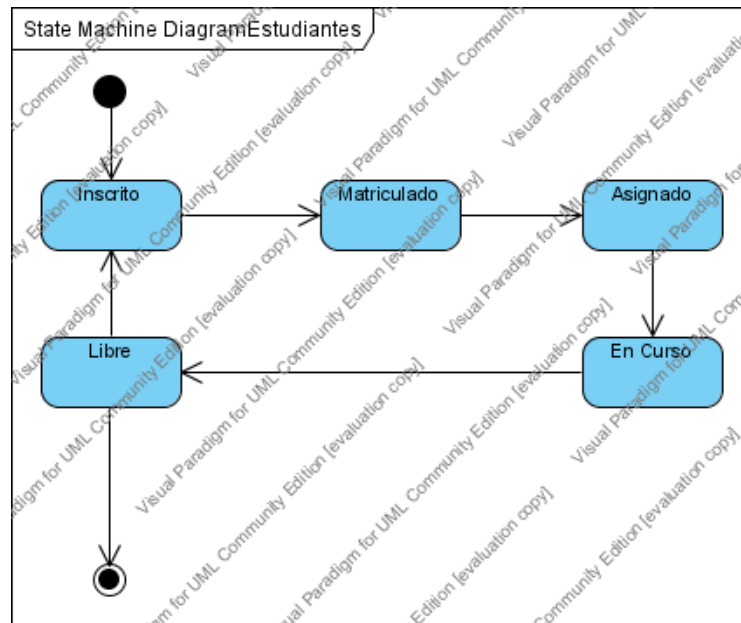
ESTADOS DE CURSO



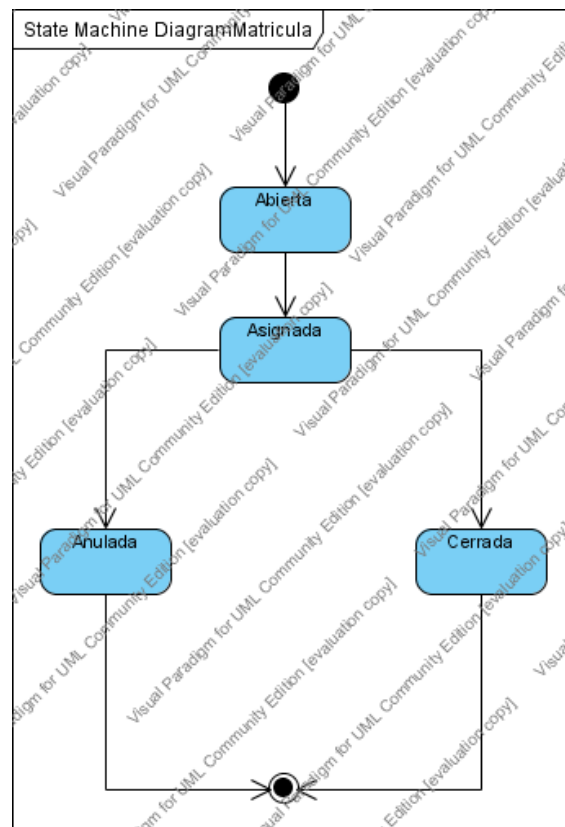
ESTADOS DE PROFESOR



ESTADOS DE ESTUDIANTES



ESTADOS DE MATRICULA



ESTADO DE DETALLES DE CURSO

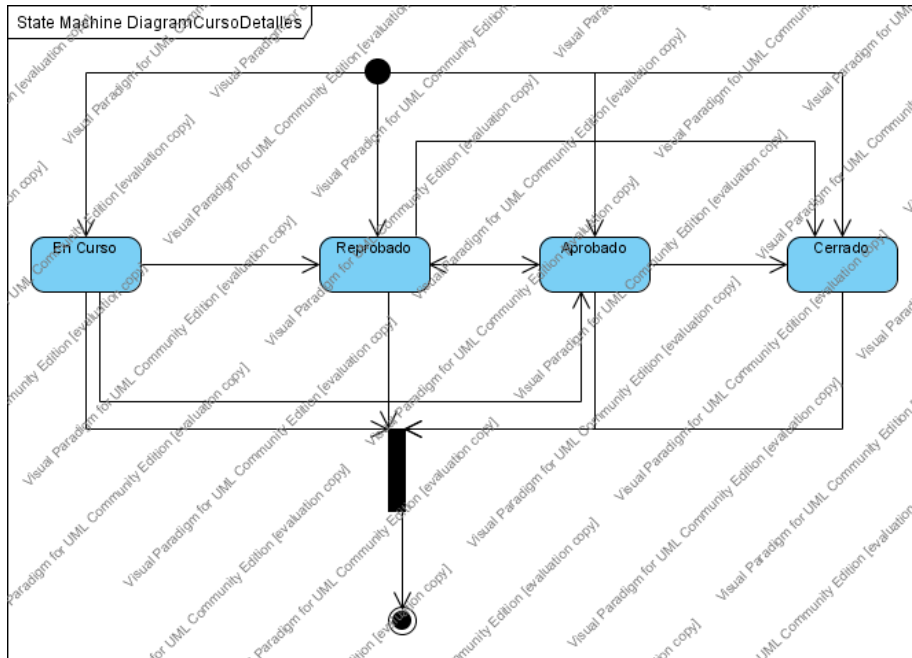


Diagrama de Paquetes

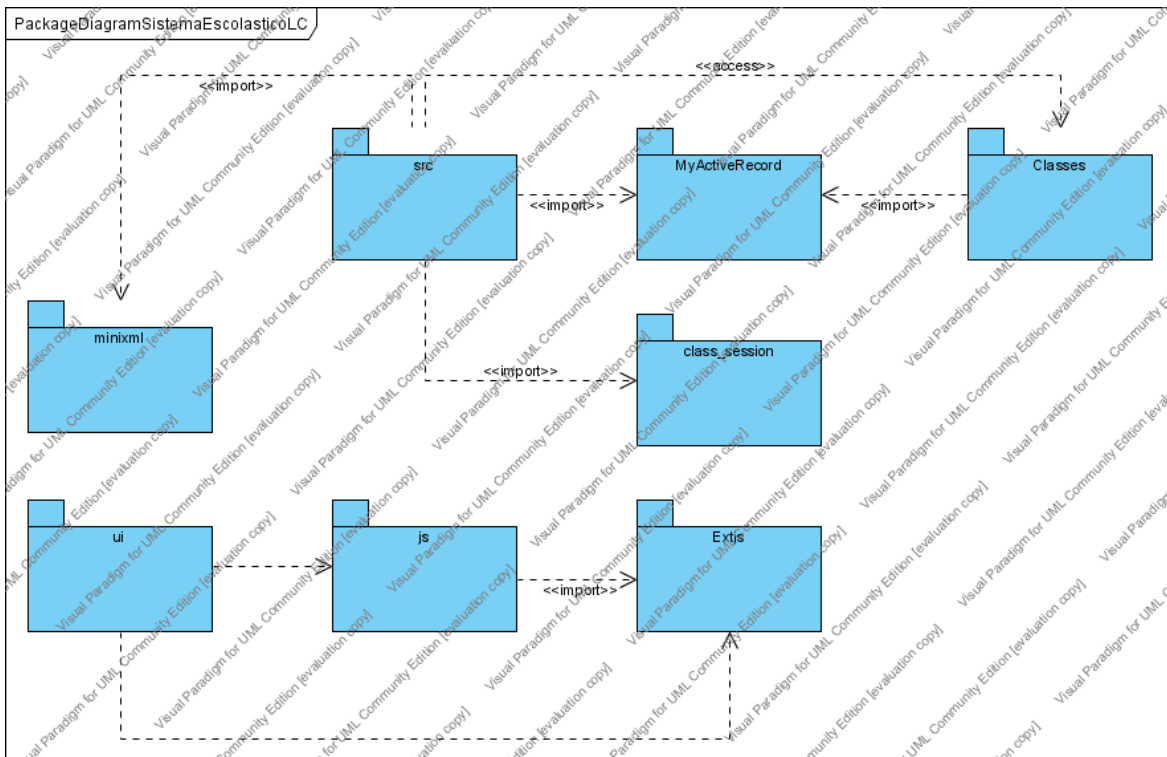
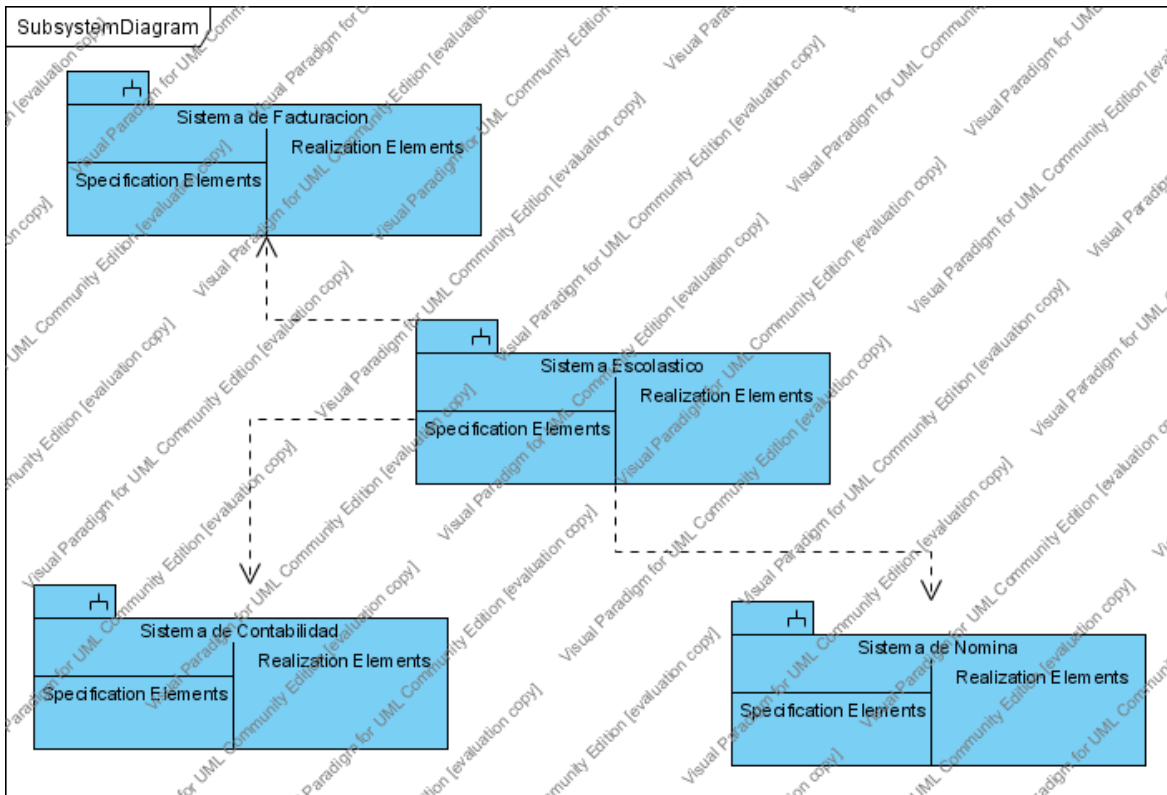


Diagrama de Subsistemas



Interfaces de Usuario (Prototipo)

VENTANA DE LOGIN

The screenshot shows a login window with a light blue background and a title bar that reads 'Lincoln College Autenticacion del Sistema'. Below the title bar, there are two input fields: the first is labeled 'Usuario:' and the second is labeled 'Password:'. At the bottom center of the window, there is a button labeled 'Login'.

Gráfico 4.1

VENTANA DE INFORMACION

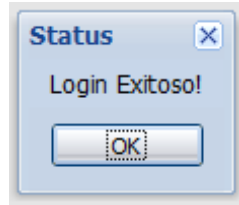


Gráfico 4.2

Gráfico 4.3 VIEWPORT ADMINISTRACION

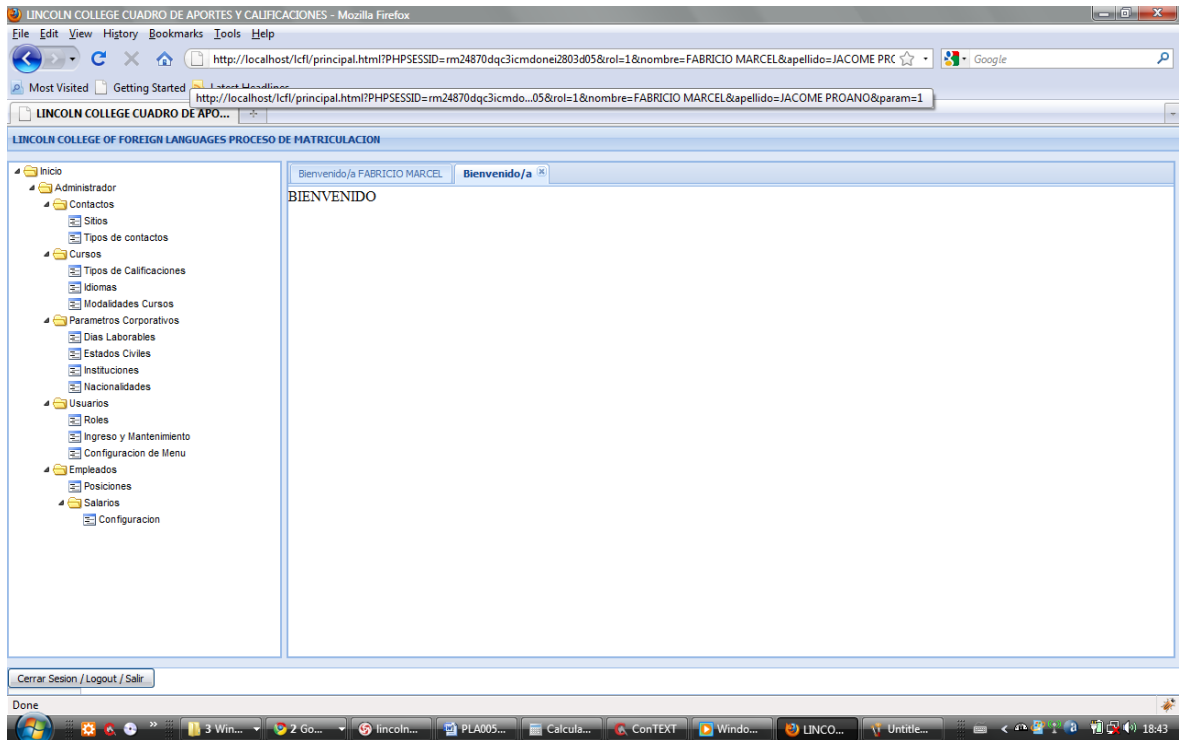


Gráfico 4.4 PARAMETROS : SITIOS

The screenshot shows the 'SITIOS' configuration page in the LINCOLN COLLEGE CUADRO DE APORTE Y CALIFICACIONES web application. The page title is 'Bienvenido/a FABRICIO MARCEL' and the page is titled 'SITIOS'. The main content area displays a table with the following data:

id	Descripcion
1	CASA
2	TRABAJO
3	ESCUELA
4	UNIVERSIDAD
5	CASA CAMPO
6	AVION PRIVADO
9	MOVIL
10	NEGOCIO
11	EXTERIOR

The table includes columns for 'Crear Sitio', 'Eliminar Sitio', and 'Refresh'. The page also shows a navigation bar with 'Cerrar Sesion / Logout / Salir' and a taskbar at the bottom with the time 18:50.

Gráfico 4.5 PARAMETROS : TIPOS DE CONTACTOS

The screenshot shows the 'TIPOS DE CONTACTOS' configuration page in the LINCOLN COLLEGE CUADRO DE APORTE Y CALIFICACIONES web application. The page title is 'Bienvenido/a FABRICIO MARCEL' and the page is titled 'TIPOS DE CONTACTOS'. The main content area displays a table with the following data:

id	Descripcion
1	TELEFONO
2	CELULAR
3	FAX
4	BEEPER
5	PAGER
6	EMAIL
7	WEBSITE

The table includes columns for 'Crear Tipo de Contacto', 'Eliminar Tipo de contacto', and 'Refresh'. The page also shows a navigation bar with 'Cerrar Sesion / Logout / Salir' and a taskbar at the bottom with the time 19:22.

Gráfico 4.6 PARAMETROS: TIPOS DE CALIFICACIONES

The screenshot shows the 'Tipos de Calificaciones' page in the LINCOLN COLLEGE CUADRO DE APORTES Y CALIFICACIONES application. The page features a navigation menu on the left and a main content area with a table of grading types. The table has columns for 'id', 'descripcion', and 'abreviacion'. The records are as follows:

id	descripcion	abreviacion
1	DEBER1	D1
2	COMPOSICION1	C1
3	PRUEBA ESCRITA1	PE1
4	PRUEBA ORAL1	PO1
5	EXAMEN ESCRITO1	EE1
6	EXAMEN ORAL1	EO1
7	PROYECTO	PR
8	EXPOSICION	EX
9	DEBER2	D2
10	COMPOSICION2	C2
11	PRUEBA ESCRITA2	PE2
12	PRUEBA ORAL2	PO2

The page also includes a 'Crear Tipo Aporte' button, an 'Eliminar Tipo Aporte' button, and a 'Refresh' button. The table is displayed on page 1 of 2, showing 13 records.

Gráfico 4.7 PARAMETROS: IDIOMAS

The screenshot shows the 'Idiomas' page in the LINCOLN COLLEGE CUADRO DE APORTES Y CALIFICACIONES application. The page features a navigation menu on the left and a main content area with a table of languages. The table has columns for 'id' and 'descripcion'. The records are as follows:

id	descripcion
1	INGLES
2	FRANCES
3	ITALIANO
4	JAPONES
5	ALEMAN
7	PORTUGUES

The page also includes a 'Crear Idioma' button and an 'Eliminar Idioma' button. The table is displayed on page 1 of 1, showing 6 records.

Gráfico 4.8 PARAMETROS: MODALIDAD DE CURSOS

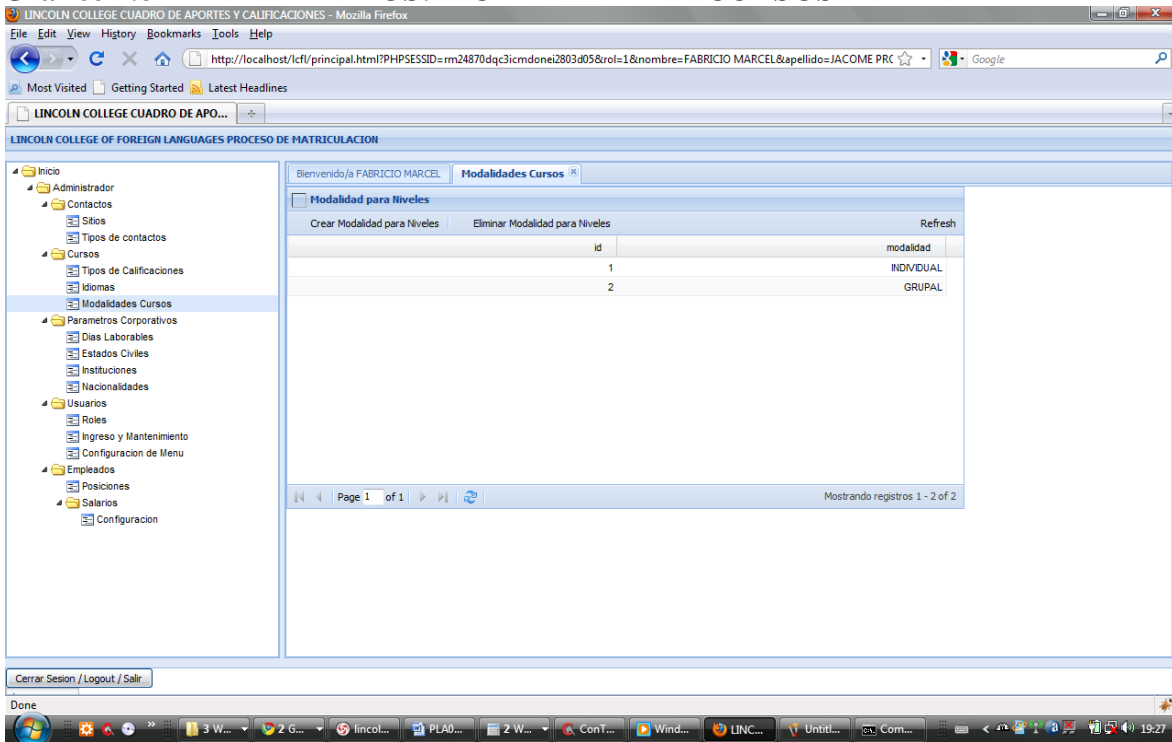


Gráfico 4.9 PARAMETROS: DIAS LABORABLES

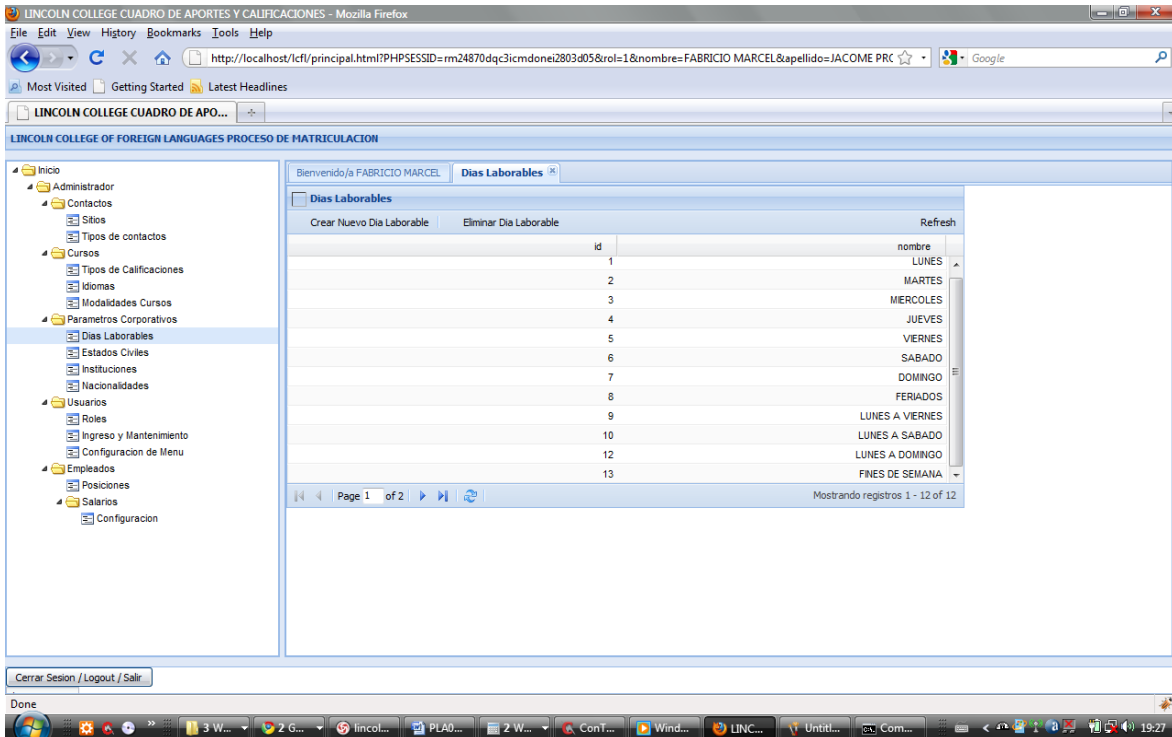


Gráfico 4.10 PARAMETROS: ESTADOS CIVILES

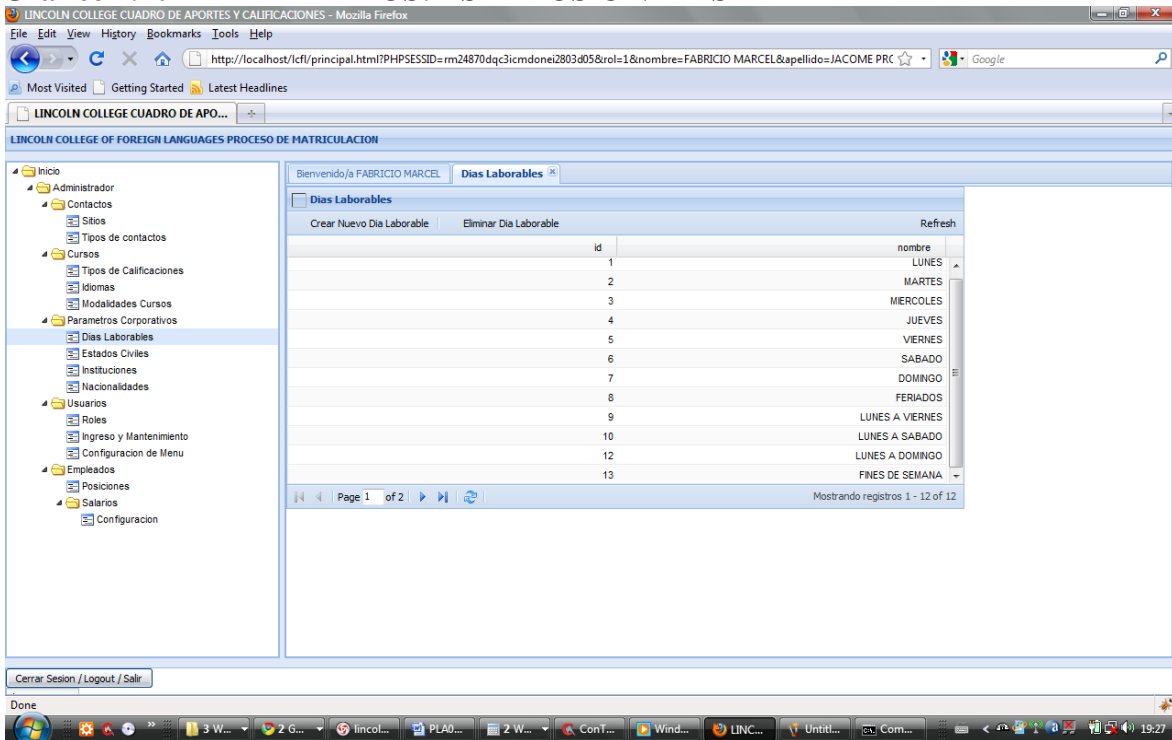


Gráfico 4.11 PARAMETROS: INSTITUCIONES

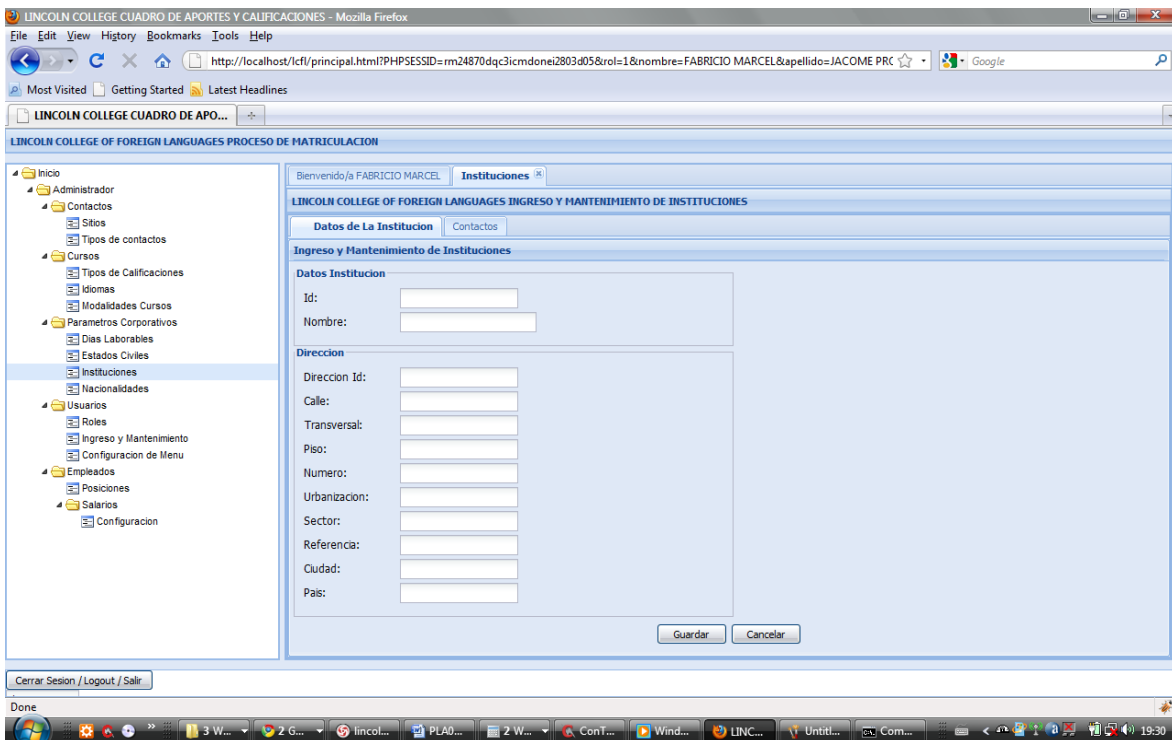


Gráfico 4.12 PANTALLA INGRESO DE INSTITUCIONES

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE INSTITUCIONES

Datos de La Institucion Contactos

Ingreso y Mantenimiento de Instituciones

Datos Institucion

Id:

Nombre:

Direccion

Direccion Id:

Calle:

Transversal:

Piso:

Numero:

Urbanizacion:

Sector:

Referencia:

Ciudad:

Pais:

Guardar Cancelar

Gráfico 4.13 PARAMETROS: NACIONALIDADES

LINCOLN COLLEGE CUADRO DE APORTES Y CALIFICACIONES - Mozilla Firefox

http://localhost/lcfi/principal.html?PHPSESSID=rm24870dq3icmdonei2803d05&rol=1&nombre=FABRICIO MARCEL&apellido=JACOME PRC

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL Nacionalidades

nacionalidad

Crear nacionalidad Eliminar nacionalidad Refresh

id	nacionalidad
1	ECUAT/CANADENSE
2	ECUATORIANA/CHILENA
3	ECUATORIANA/CHINA
4	ECUATCOLOMBIANA
5	ECUAT/CUBANA
6	ECUAT/ESPAÑOLA
7	ECUAT/ESTADOUNIDENSE
8	ECUAT/FLANDESA
9	ECUAT/FRANCESA
10	ECUAT/MEXICANA
11	ECUAT/PANAMENA
12	ECUAT/PERUANA

Page 1 of 23 Mostrando registros 1 - 227 of 227

Cerrar Sesión / Logout / Salir

Gráfico 4.14 PARAMETROS: ROLES DE USUARIOS

Lincoln College Cuadro de Aportes y Calificaciones - Mozilla Firefox

http://localhost/lcfl/principal.html?PHPSESSID=rm24870dq3icmdonei2803d05&rol=1&nombre=FABRICIO MARCEL&apellido=JACOME PRC

LINCOLN COLLEGE CUADRO DE APO... LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL Roles

Roles de Usuario

Crear Roles de Usuario Eliminar Roles de Usuario Refresh

id	rol
1	Administrador
2	Operador
3	Profesor
5	Alumno
6	Empleado
17	prueba

Page 1 of 1 Mostrando registros 1 - 6 of 6

Cerrar Sesión / Logout / Salir

Gráfico 4.15 PARAMETROS: USUARIOS

Lincoln College Cuadro de Aportes y Calificaciones - Mozilla Firefox

http://localhost/lcfl/principal.html?PHPSESSID=rm24870dq3icmdonei2803d05&rol=1&nombre=FABRICIO MARCEL&apellido=JACOME PRC

LINCOLN COLLEGE CUADRO DE APO... LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL Ingreso y Mantenimiento

Cuentas de Usuarios del Sistema

Crear un nuevo usuario Eliminar Usuarios Buscar Imprimir

#	Id Persona	Usuario	Fecha Ingreso	Fecha Modificación	Rol
1	1	gcarrillo	2008-03-28 00:00:00	2009-10-04 05:32:50	3
2	2	fjacome	2008-03-10 00:00:00	2008-05-09 00:00:00	1
3	3	ajacome	2008-04-12 00:00:00	2008-10-15 00:00:00	2
5	4	jip	2008-05-09 00:00:00	2008-09-04 00:00:00	1
7	4	jiplan	2008-05-10 00:00:00	2009-07-02 17:47:05	2
8	5	lzapatlas	2008-10-17 00:00:00	2009-05-08 00:00:00	2
9	7	lipo	2008-10-17 00:00:00	2008-10-17 00:00:00	5
10	8	usu	2008-10-17 00:00:00	2009-05-08 00:00:00	3
12	30	hshash2	2009-05-08 22:31:05	2009-05-08 22:33:50	2
13	2	fjacome	2009-08-04 01:10:29	2009-11-14 19:13:51	3
14	2	mfbr1	2009-08-30 08:22:06	0000-00-00 00:00:00	2

Page 1 of 1 Displaying 1 - 11 of 11

Cerrar Sesión / Logout / Salir

Gráfico 4.16 PANTALLA CONSULTA USUARIOS DEL SISTEMA

Cuentas de Usuarios del Sistema						
Crear un nuevo usuario Eliminar Usuarios Buscar Imprimir						
#	Id Persona	Usuario	Fecha Ingreso	Fecha Modificacion	Rol	
1	1	gcarrillo	2008-03-28 00:00:00	2009-10-04 05:32:50	3	
2	2	fjacome	2008-03-10 00:00:00	2008-05-09 00:00:00	1	
3	3	ajacome	2008-04-12 00:00:00	2008-10-15 00:00:00	2	
5	4	jtp	2008-05-09 00:00:00	2008-09-04 00:00:00	1	
7	4	jtipan	2008-05-10 00:00:00	2009-07-02 17:47:05	2	
8	5	lzapatas	2008-10-17 00:00:00	2009-05-08 00:00:00	2	
9	7	lipo	2008-10-17 00:00:00	2008-10-17 00:00:00	5	
10	8	usu	2008-10-17 00:00:00	2009-05-08 00:00:00	3	
12	30	hshash2	2009-05-08 22:31:05	2009-05-08 22:33:50	2	
13	2	fjacome	2009-08-04 01:10:29	2009-11-14 19:13:51	3	
14	2	rnfbr1	2009-08-30 08:22:06	0000-00-00 00:00:00	2	

Page 1 of 1 | Displaying 1 - 11 of 11

Gráfico 4.17 PARAMETROS: CONFIGURACION DE OPCIONES DE MENU POR ROL

Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/lcf/administracionMenus.htm

LINCOLN COLLEGE OF FOREIGN LANGUAGES CONFIGURACION DE OPCIONES DE MENU

Escoger un Rol de Usuario

Rol de Usuario: Administrador

Cancelar

OPCIONES DE MENU NIVEL RAIZ

Nueva Opcion	Actualizar Opcion	Eliminar Opcion			
#	Padre Id	Texto Menu	Tipo	Uri	Href
43	39	Contactos	folder		
44	39	Cursos	folder		
45	39	Parametros Corporativos	folder		
46	39	Usuarios	folder		

Page 1 of 1 | Displaying 1 - 8 of 8

OPCIONES DE MENU NIVEL UNO

Nueva Opcion	Actualizar Opcion	Eliminar Opcion			
#	Padre Id	Texto Menu	Tipo	Uri	Href
95	90	Temas	leaf	dssssss	
97	90	Temas 1	folder	dddd	sssssss
98	90	Temas 2	folder		
99	90	Temas 3	folder		

Page 1 of 1 | Displaying 1 - 4 of 4

OPCIONES DE MENU NIVEL DOS

Nueva Opcion	Actualizar Opcion	Eliminar Opcion			
#	Padre Id	Texto Menu	Tipo	Uri	Href
101	99	Temas 31	leaf		
115	99	Temas 32	leaf	sadfdsfasfd	

Page 1 of 1 | Displaying 1 - 2 of 2

Done

3 W... 2 G... linc... PLA0... 2 W... ConT... Wind... Mozil... Unitt... Cu. Com... 19:36

Gráfico 4.18 PARAMETROS: POSICIONES

Table: Posiciones

id	posicion
1	PROFESOR
2	ASISTENTE DE PROFESOR
3	SUSTITUTO

Gráfico 4.19 PARAMETROS: CONFIGURACION DE SALARIOS

Table: Datos de Posiciones Empleados

#	id	Posicion	Tipo Posicion	Salario	Categoria
5	1	PROFESOR	HORAS 2	4.50	H
6	1	PROFESOR	HORAS 3	6.25	H
1	1	PROFESOR	TIEMPO COMPLETO 1	350.00	TC
2	1	PROFESOR	TIEMPO COMPLETO 2	500.00	TC
3	1	PROFESOR	TIEMPO COMPLETO 3	900.00	TC
10	2	ASISTENTE DE PROFESOR	HORAS 1	2.25	H
11	2	ASISTENTE DE PROFESOR	HORAS 2	3.00	H
12	2	ASISTENTE DE PROFESOR	HORAS 3	3.50	H
18	2	ASISTENTE DE PROFESOR	HORAS 4	2.00	H
7	2	ASISTENTE DE PROFESOR	TIEMPO COMPLETO 1	150.00	TC
8	2	ASISTENTE DE PROFESOR	TIEMPO COMPLETO 2	250.00	TC

Gráfico 4.20 VIEWPORT OPERADOR

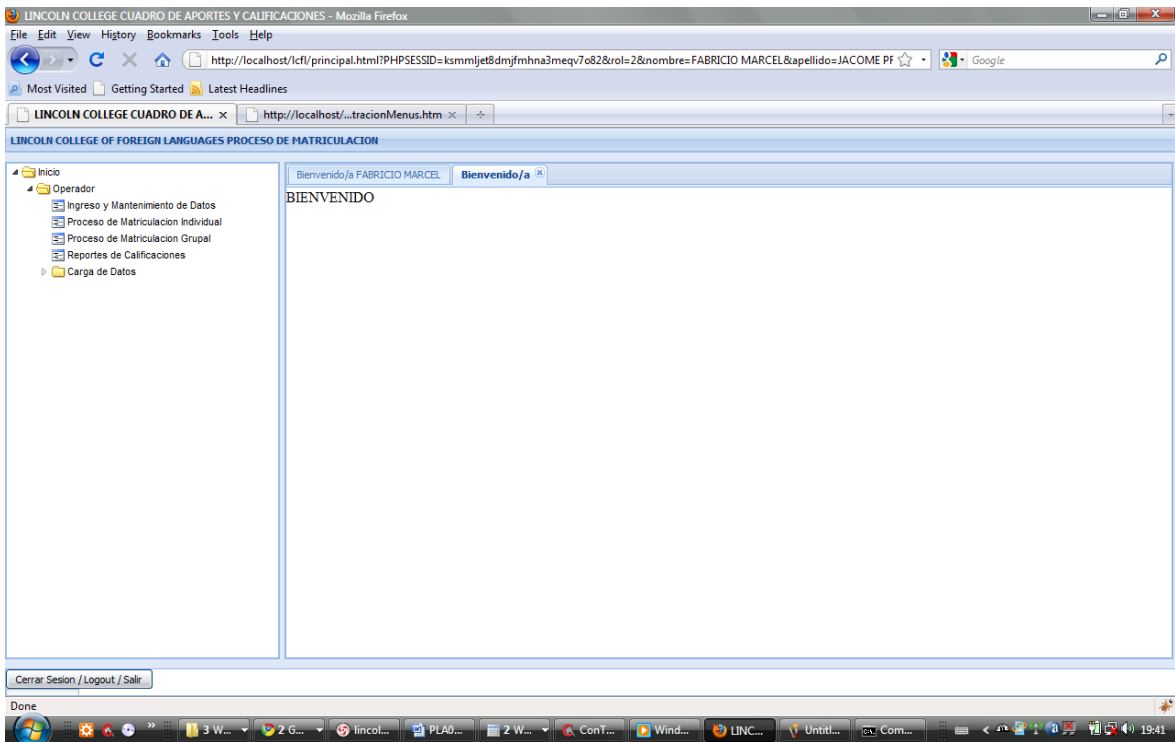


Gráfico 4.21 INGRESO Y MANTENIMIENTO DE DATOS

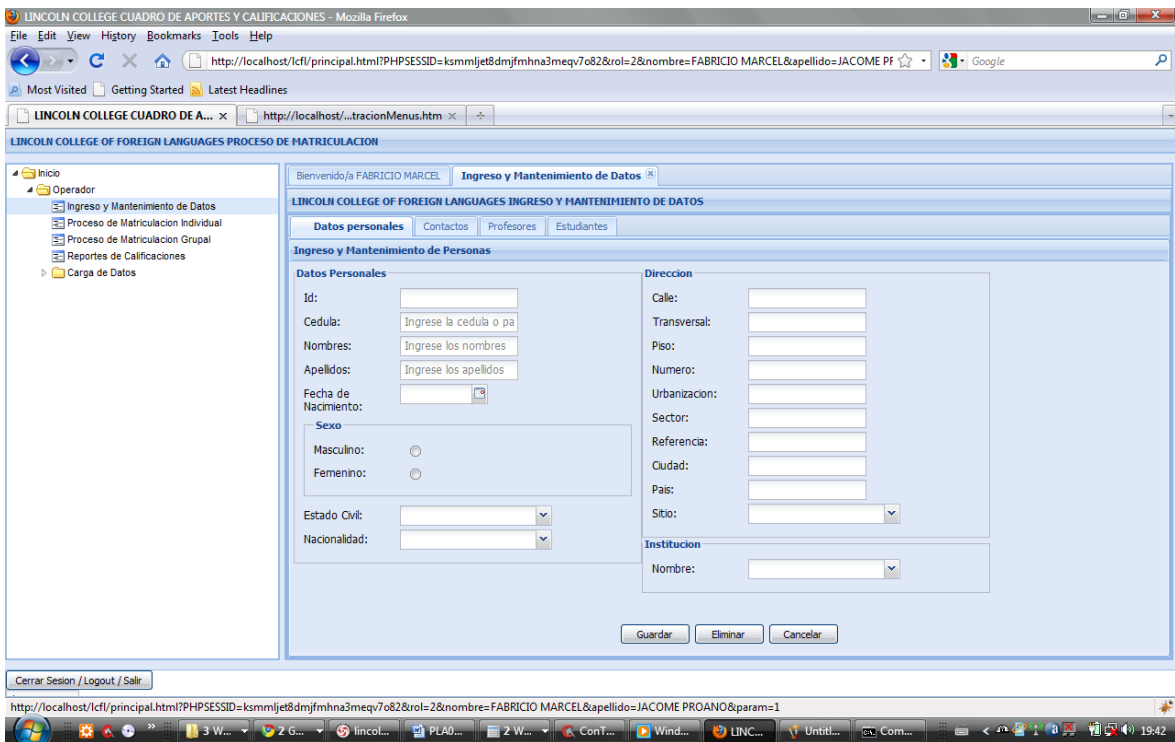


Gráfico 4.22 INGRESO Y MANTENIMIENTO DE DATOS: DATOS PERSONALES

Bienvenido/a FABRICIO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales | Contactos | Profesores | Estudiantes

Ingreso y Mantenimiento de Personas

Datos Personales

Id: 22
 Cedula: 1313131313
 Nombres: JOSE LUIS
 Apellidos: GARCIA ARCALLE
 Fecha de Nacimiento: 10/08/1981
 Sexo:
 Masculino:
 Femenino:
 Estado Civil: CASADO
 Nacionalidad: ECUATORIANA

Direccion

Calle: TERAN FERNANDEZ
 Transversal: AV. FLORIDA
 Piso: 1
 Numero: 142
 Urbanizacion: LA FLORIDA
 Sector: EL AEROPUERTO
 Referencia: CASA DE DOS PISOS
 Ciudad: QUITO
 Pais: ECUADOR
 Sitio: CASA

Institucion

Nombre: Lan Chile

Guardar Eliminar Cancelar

Gráfico 4.23 INGRESO Y MANTENIMIENTO DE DATOS: CONTACTOS

Bienvenido/a FABRICIO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales | **Contactos** | Profesores | Estudiantes

Crear un nuevo contacto Eliminar Contactos

Sitio	Tipo Contacto	Valor	Fecha Ingreso	Fecha Modificacion
CASA	TELEFONO	2765687	2008-10-25 00:00:00	0000-00-00 00:00:00

Page 1 of 1 | Displaying 1 - 1 of 1

Gráfico 4.24 INGRESO Y MANTENIMIENTO DE DATOS: NUEVO CONTACTO

BIENVENIDO/A FABRICIO MARCEL | Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales | **Contactos** | Profesores | Estudiantes

Crear un nuevo contacto | Eliminar Contactos

Sitio	Tipo Contacto	Valor	Fecha Ingreso	Fecha Modificacion
CASA	TELEFONO	2785687	2008-10-25 00:00:00	0000-00-00 00:00:00

Crear nuevo Contacto

Sitio: TRABAJO

Tipo Contacto: TELEFONO

Valor: 2433531

Guardar Cancelar

Page 1 of 1 | Displaying 1 - 1 of 1

Gráfico 4.25 INGRESO Y MANTENIMIENTO DE DATOS: PROFESORES

BIENVENIDO/A FABRICIO MARCEL | Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales | Contactos | **Profesores** | Estudiantes

DATOS DE EMPLEADO

Posicion

Empleado Id: 12

Profesor Id: 19

Estado Id: activo

Crear Empleado Activar Profesor Desactivar Profesor

IDIOMAS +

DISPONIBILIDAD +

INFORMACION SALARIAL +

Gráfico 4.26 INGRESO Y MANTENIMIENTO DE DATOS: PROFESORES – IDIOMAS

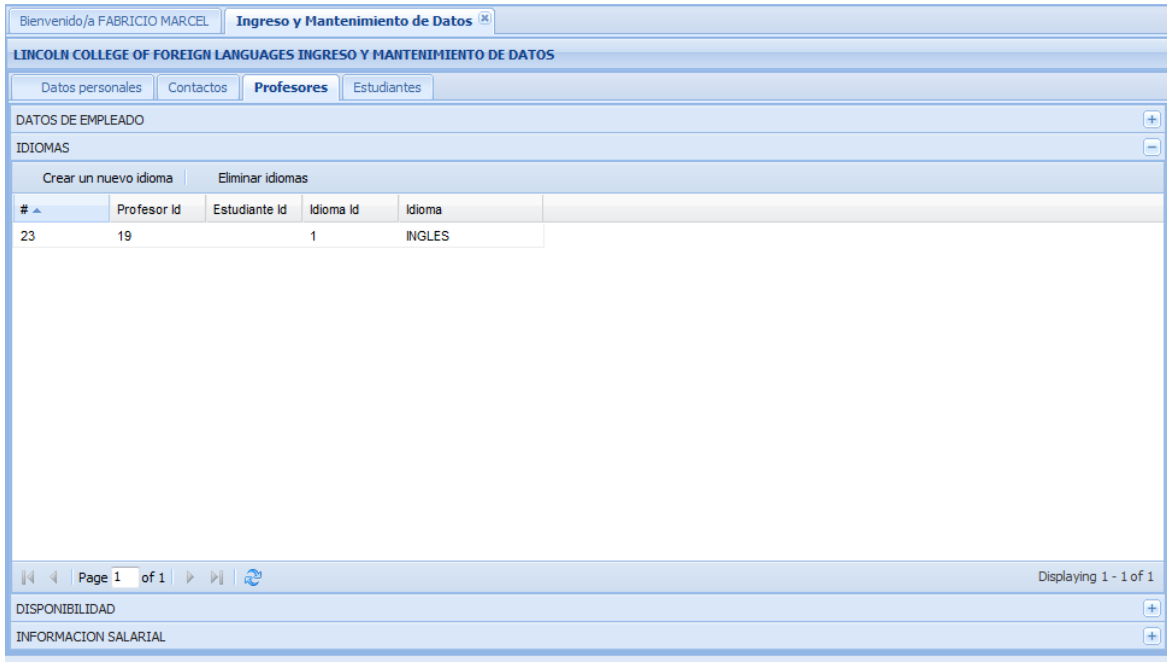


Gráfico 4.27 CREAR NUEVO IDIOMA PARA PROFESOR

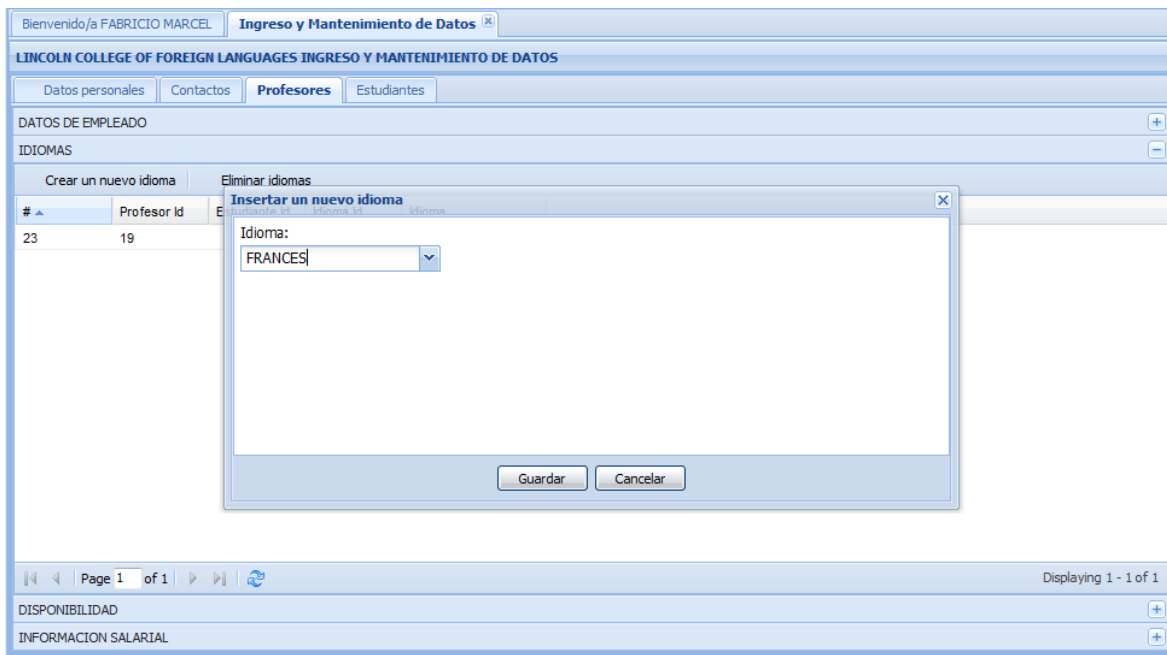


Gráfico 4.28 INGRESO Y MANTENIMIENTO DE DATOS: PROFESOR – DISPONIBILIDAD

The screenshot shows a web browser window displaying the 'LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION' application. The user is logged in as FABRICIO MARCEL. The main content area is titled 'INGRESO Y MANTENIMIENTO DE DATOS' and features a sidebar menu on the left with options like 'Inicio', 'Operador', and 'Ingreso y Mantenimiento de Datos'. The main area has tabs for 'Datos personales', 'Contactos', 'Profesores', and 'Estudiantes'. Below these tabs, there are sections for 'DATOS DE EMPLEADO', 'IDIOMAS', and 'DISPONIBILIDAD'. The 'DISPONIBILIDAD' section contains a table with the following data:

#	Profesor Id	Idioma	Dia Disponible	Fecha Ingreso	Fecha Modificacion	Comienzo	Fin
40	19	INGLES	LUNES	2009-11-21 20:12:31	0000-00-00 00:00:00	7:00 AM	8:00 AM

At the bottom of the page, there is a footer with 'Page 1 of 1' and 'Displaying 1 - 1 of 1'. The taskbar at the bottom shows several open applications and the system clock at 20:12.

Gráfico 4.29 CREAR NUEVA DISPONIBILIDAD

The screenshot shows the same web application as in Gráfico 4.28, but with a modal dialog box titled 'Crear nueva Disponibilidad' open. The dialog box contains the following form fields:

- Idioma: INGLES
- Dia Disponible: LUNES
- Hora Inicio: 9:00 AM
- Hora Fin: 10:00 AM

At the bottom of the dialog box, there are two buttons: 'Guardar' and 'Cancelar'. The background page is dimmed, and the footer shows 'Page 1 of 1' and 'Displaying 1 - 2 of 2'.

Gráfico 4.30 INGRESO Y MANTENIMIENTO DE DATOS: PROFESORES – INFORMACION SALARIAL

The screenshot shows a web application interface for 'LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS'. The user is logged in as 'FABRICIO MARCEL'. The main menu includes 'Datos personales', 'Contactos', 'Profesores', and 'Estudiantes'. The 'Profesores' section is active, showing a list of salary records. The 'INFORMACION SALARIAL' section is expanded, showing a table with columns 'Posicion', 'Tipo Posicion', and 'Salario'. A single record is displayed: 'PROFESOR' with 'TIEMPO COMPLETO 1' and a salary of '350.00'. The interface also includes a 'Crear salario' button and a 'Eliminar salario' button. The footer indicates 'Page 1 of 1' and 'Displaying 1 - 1 of 1'.

Posicion	Tipo Posicion	Salario
PROFESOR	TIEMPO COMPLETO 1	350.00

Gráfico 4.31 CREACION DE SALARIO

The screenshot shows the same web application interface as in Gráfico 4.30, but with a 'Crear nuevo Salario' dialog box open. The dialog box has two dropdown menus: 'Posicion' set to 'ASISTENTE DE PROFESO' and 'Descripcion' set to 'TIEMPO COMPLETO 2'. At the bottom of the dialog box are 'Guardar' and 'Cancelar' buttons. The background interface is dimmed, showing the 'Profesores' section and the 'INFORMACION SALARIAL' section. The footer indicates 'Page 1 of 1' and 'Displaying 1 - 1 of 1'.

Gráfico 4.32 MATRICULACION

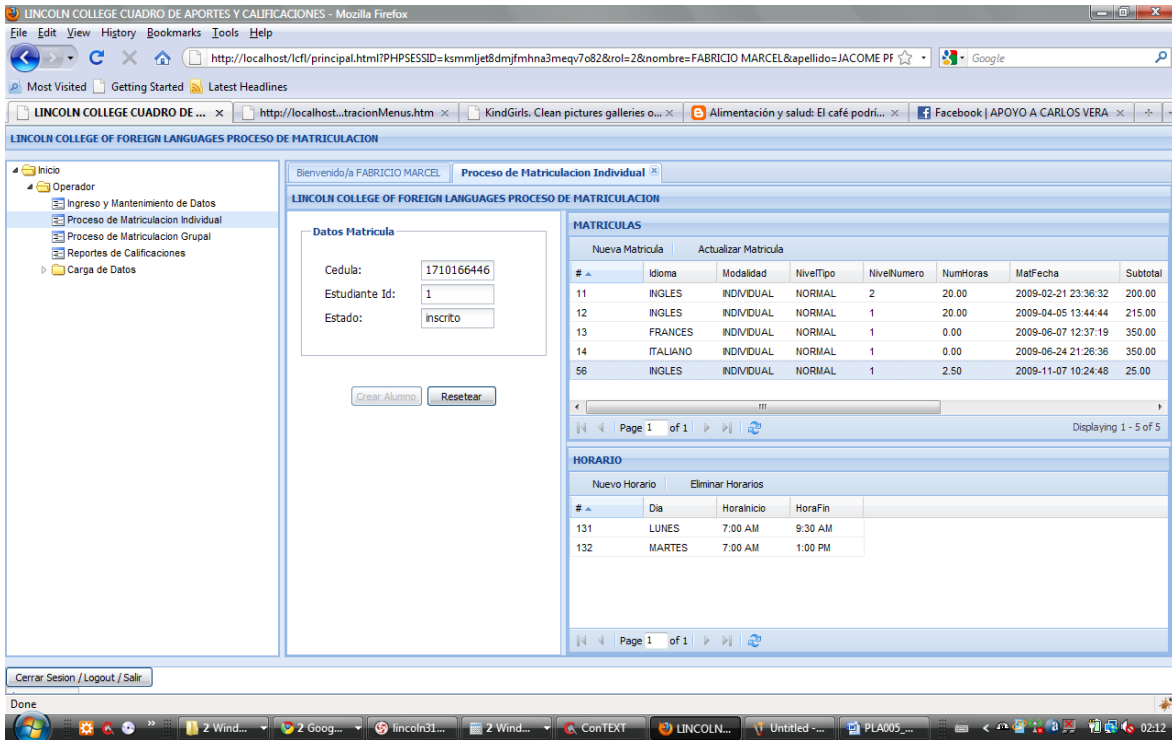


Gráfico 4.33 CREAR NUEVA MATRICULA

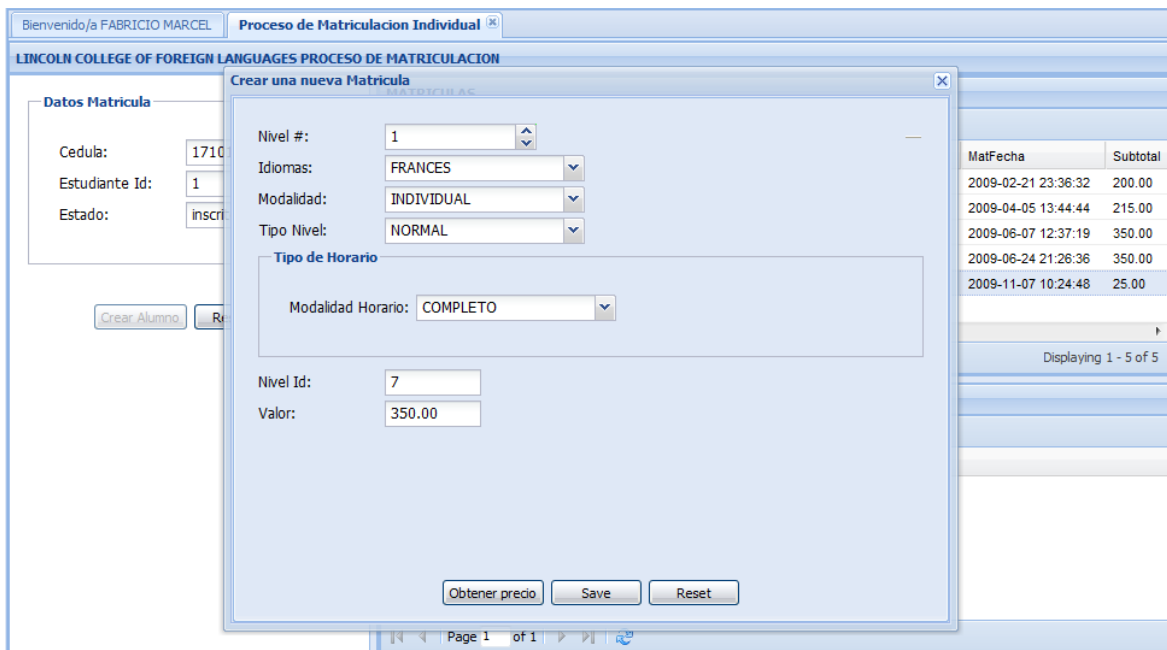


Gráfico 4.34 ASIGNACION DE PROFESOR / CREACION CURSO

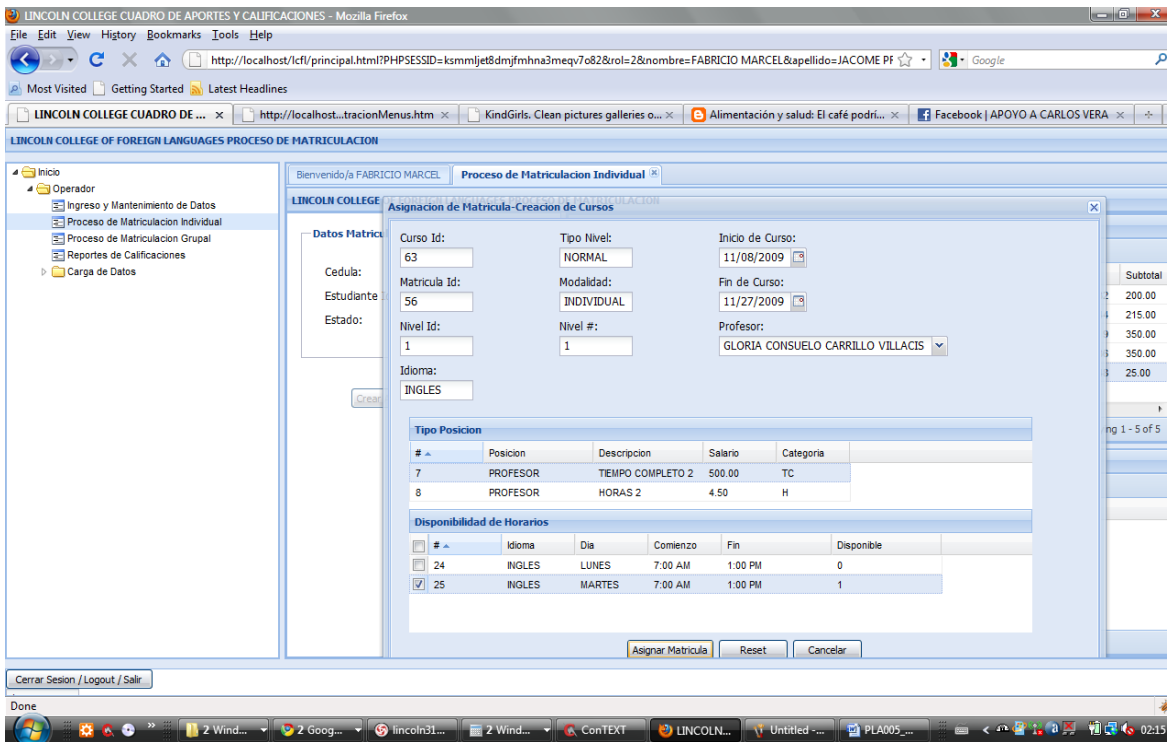


Gráfico 4.35 MATRICULACION GRUPAL

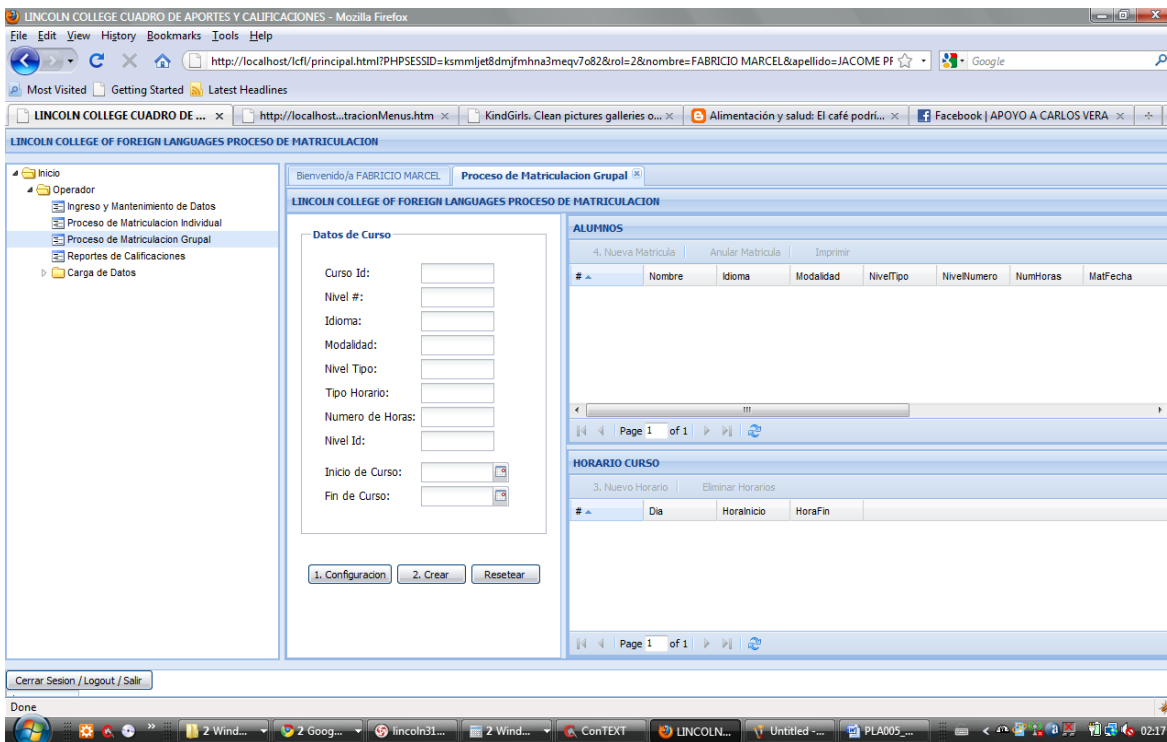


Gráfico 4.36 INGRESO CONFIGURACION DE CURSO

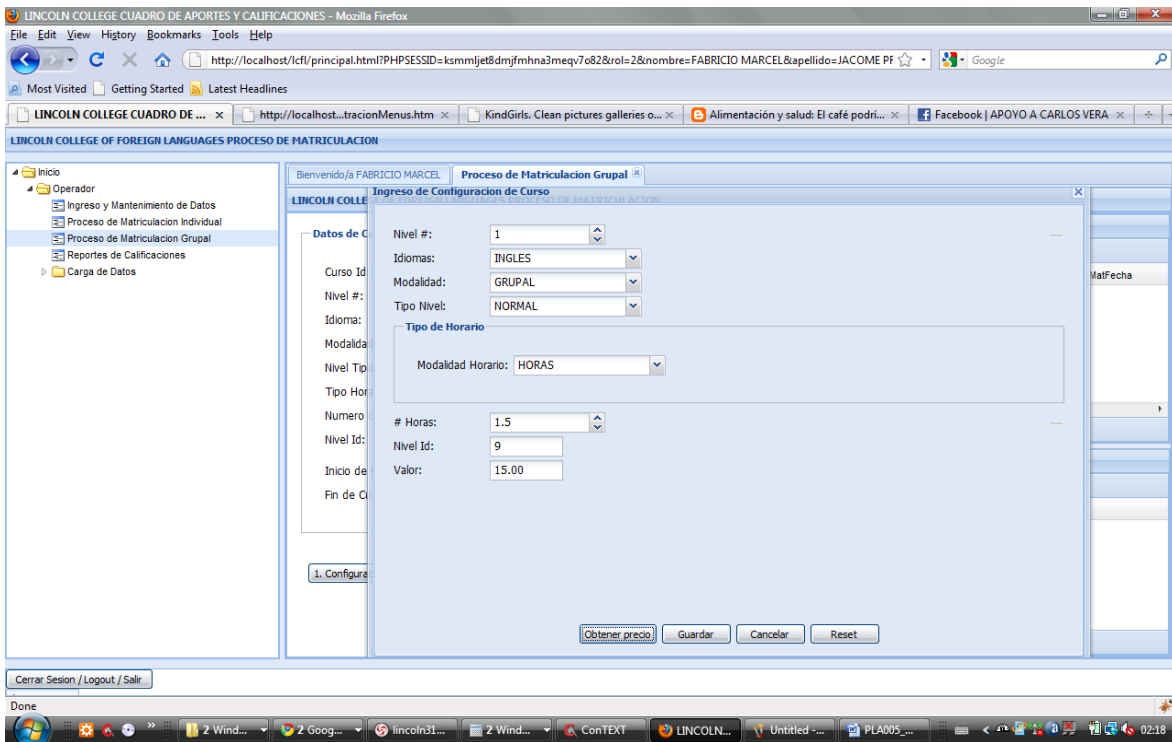


Gráfico 4.37 CREACION DE CURSO / ASIGNACION DE HORARIO / ASIGNACION DE PROFESOR

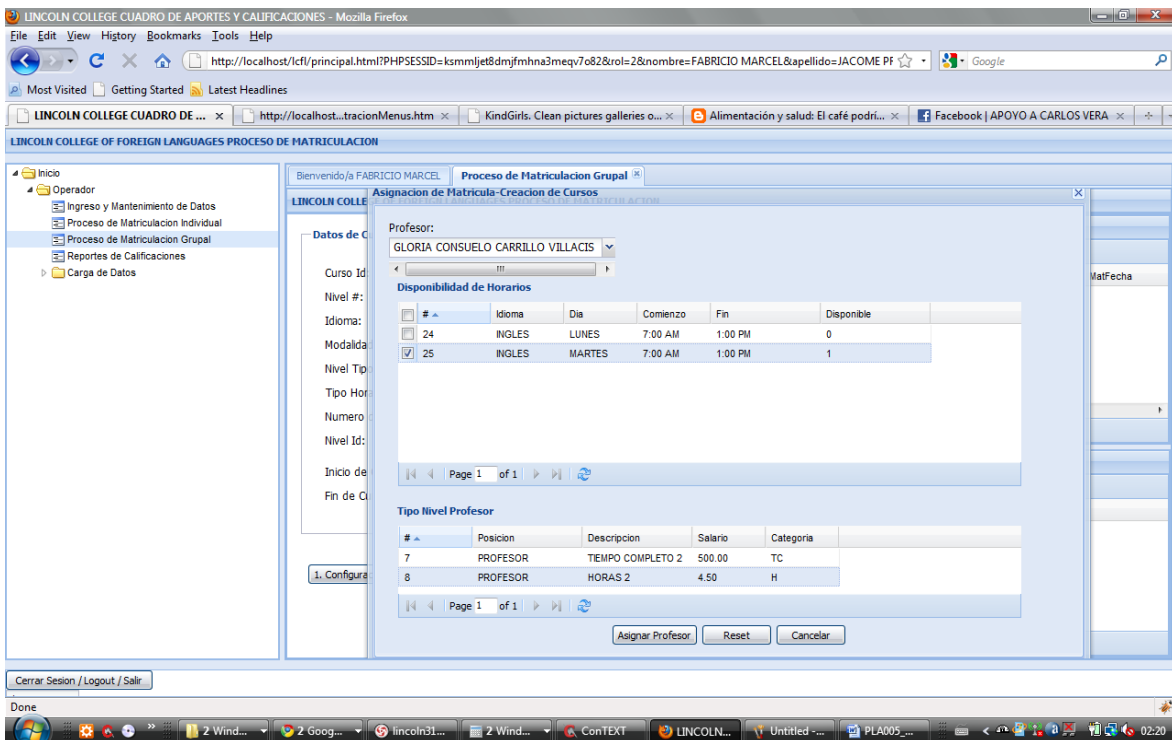


Gráfico 4.38 CREACION MATRICULA

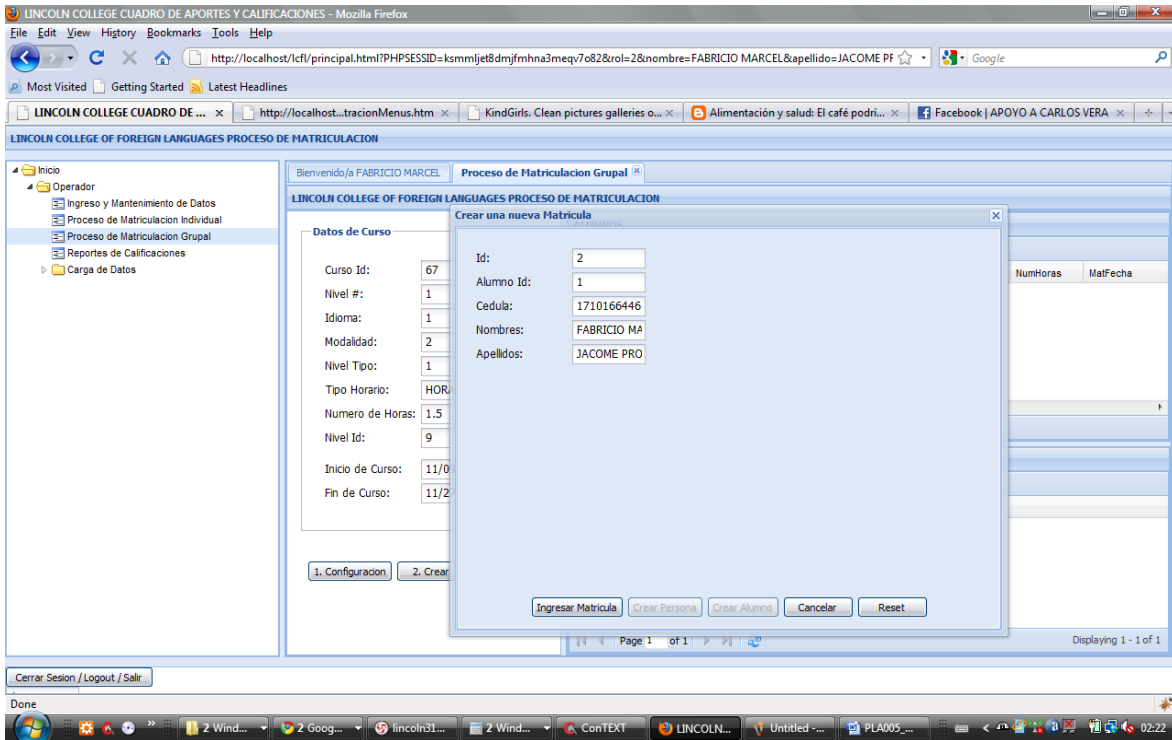


Gráfico 4.39 VIEWPORT PROFESOR

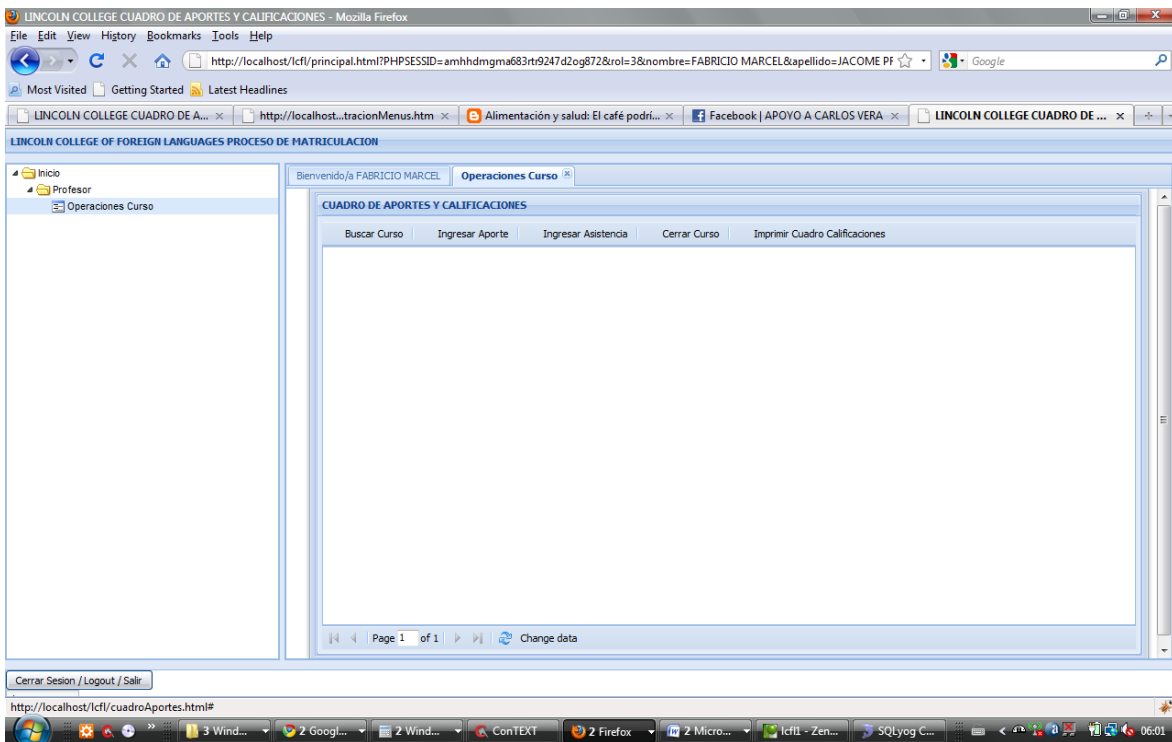


Gráfico 4.40 BUSQUEDA DE CURSO

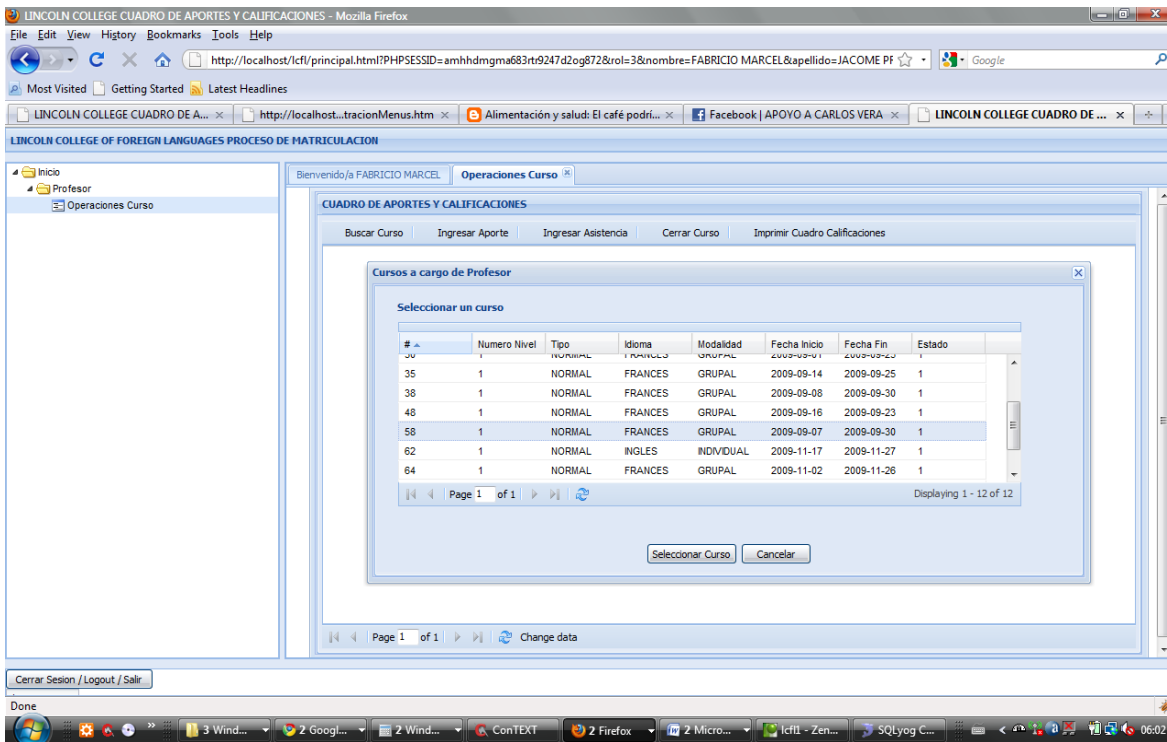


Gráfico 4.41 CUADRO DE CALIFICACIONES

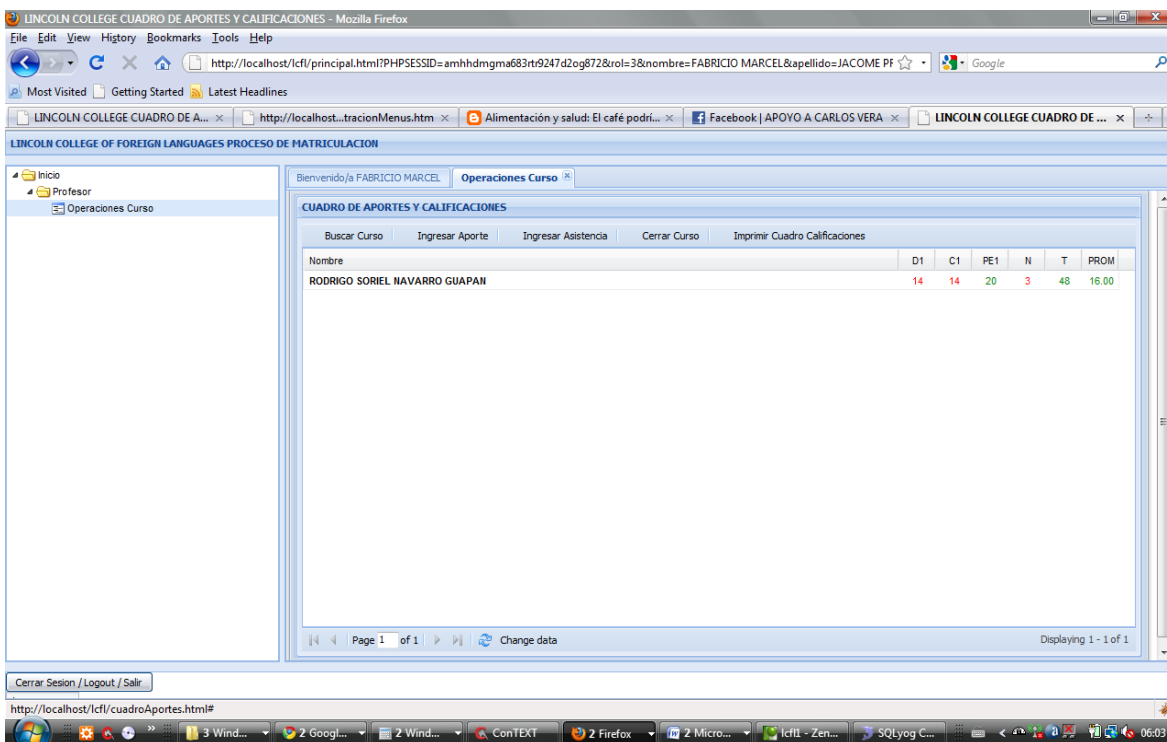


Gráfico 4.42 INGRESO DE APORTES

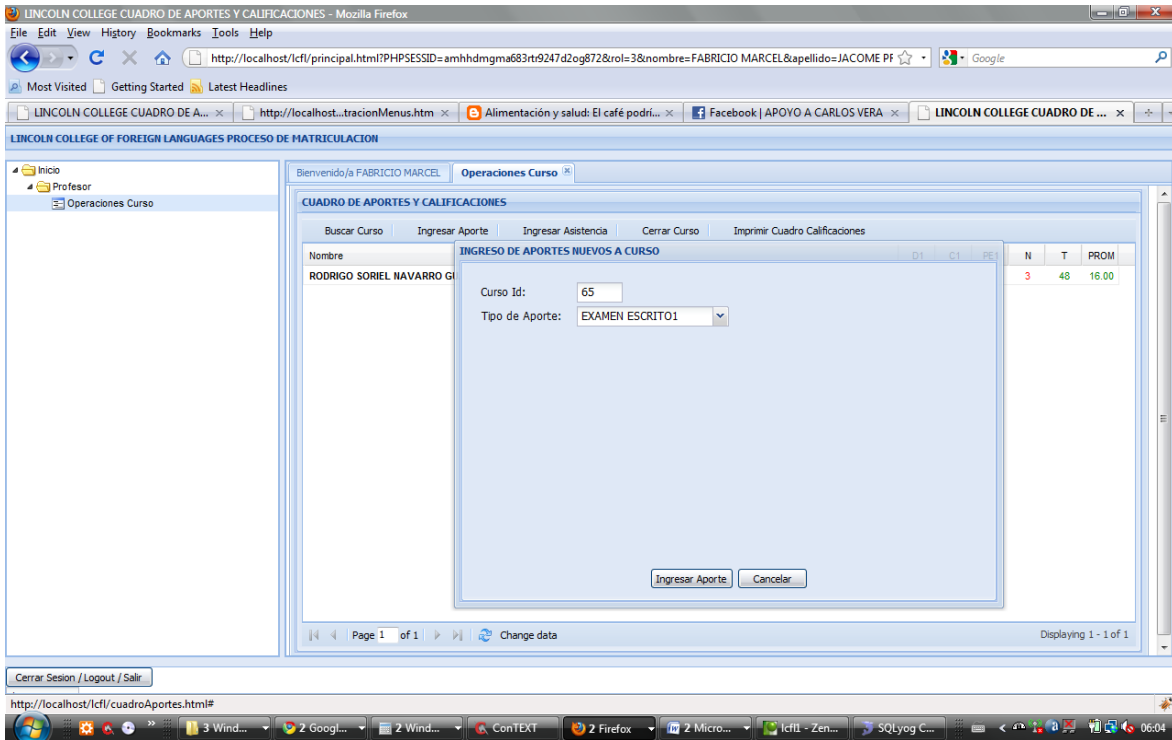


Gráfico 4.43 INGRESO DE ASISTENCIA

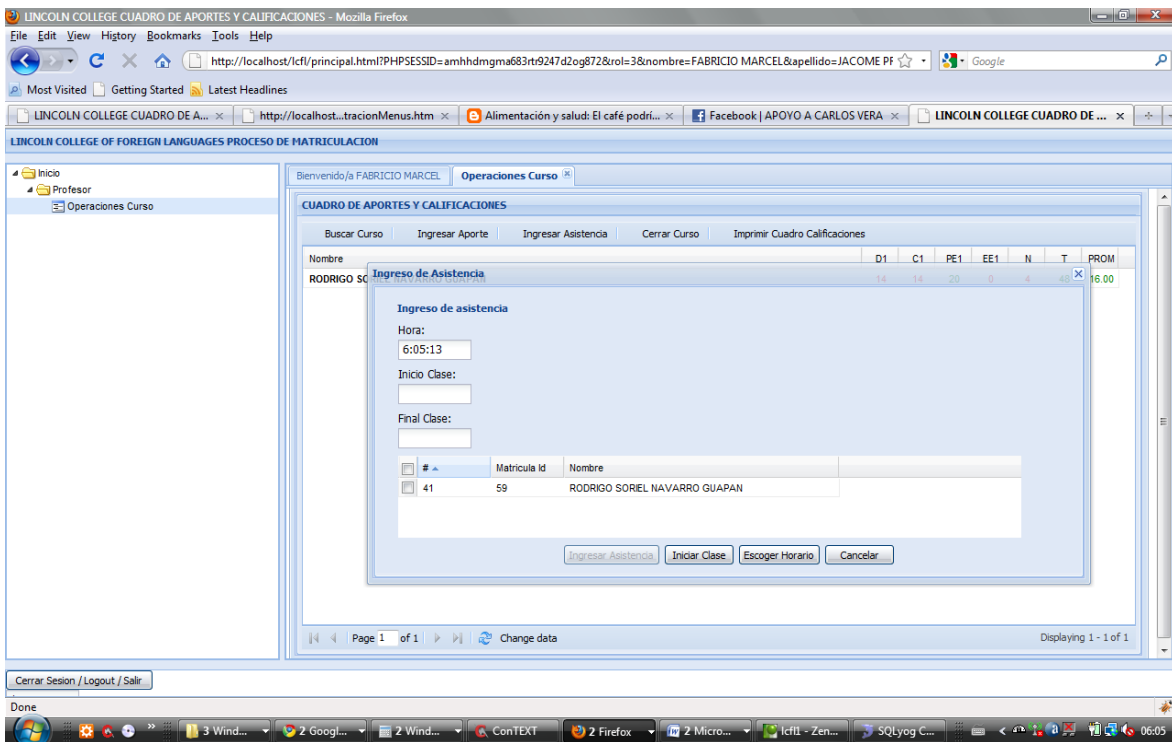


Gráfico 4.44 SELECCIONAR HORARIO DEL CURSO A INGRESAR ASISTENCIA

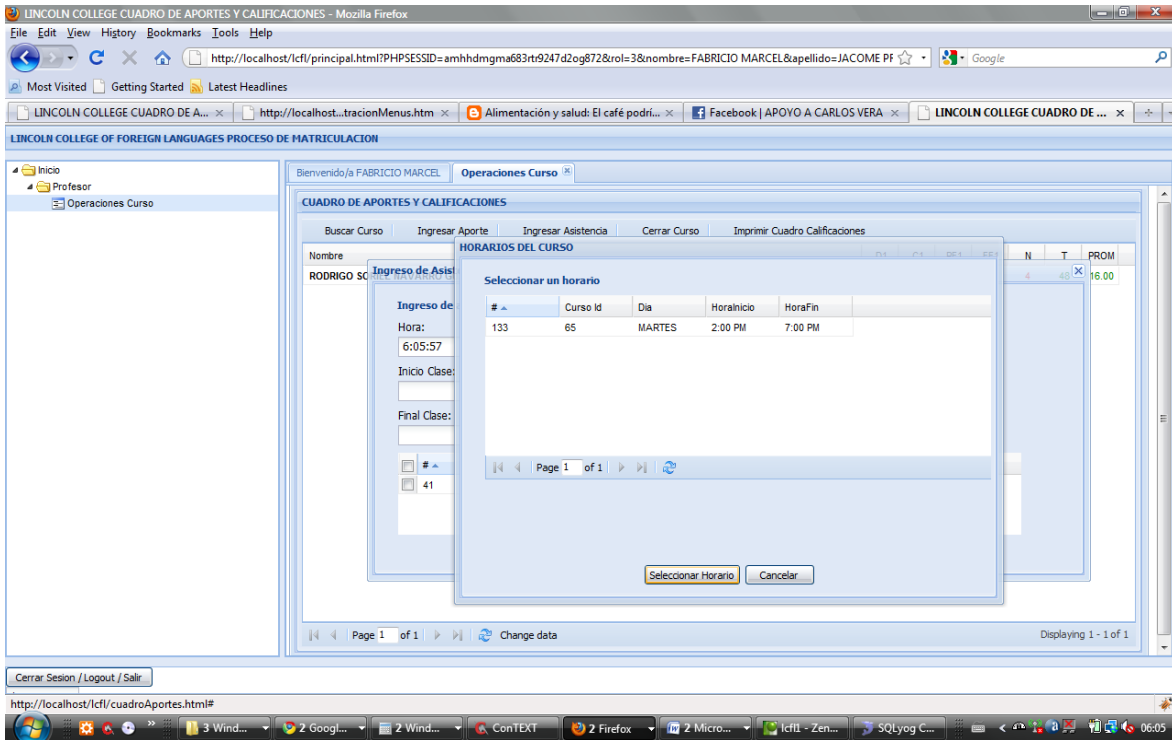
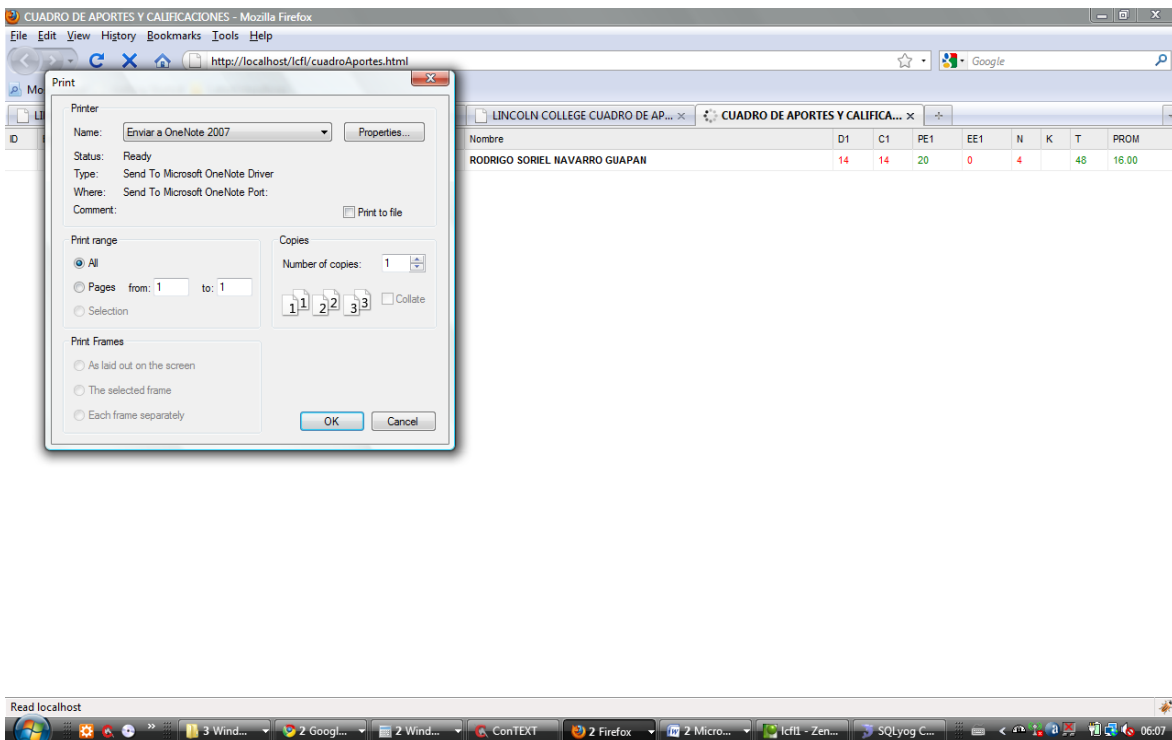


Gráfico 4.45 IMPRIMIR CUADRO DE CALIFICACIONES



4.5 FASE DE CONSTRUCCION

Una vez desarrollado el proyecto, se presentan los casos de prueba con los resultados debidamente documentados. El código fuente, al ser extenso se incluye en un medio magnético.

4.5.1 CASOS DE PRUEBA

Los casos de prueba son artefactos derivan de los casos de uso del sistema realizados en la fase de elaboración.

FASE: CONSTRUCCIÓN	ID: CNS001
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de Prueba: Ingresar y Mantener Datos
VERSIÓN:	1.0

Descripción

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ingreso y Mantenimiento de Datos”.

Se ingresa datos de una persona a la aplicación mediante el formulario de Ingreso de datos con el perfil de Operador.

Prueba 1: Comprobar que el sistema recupera los datos de una persona correctamente.

Descripción

Ingresamos al sistema como Operador , accediendo a la funcionalidad de Ingreso y Mantenimiento de Datos .

Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el sistema busque una cedula existente y recupere los datos de la persona.

Entrada

- Introducimos la cedula “1717732505” en el campo Cédula

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales Contactos Profesores Estudiantes

Ingreso y Mantenimiento de Personas

Datos Personales

Id:

Cedula:

Nombres:

Apellidos:

Fecha de Nacimiento:

Sexo

Masculino:

Femenino:

Estado Civil:

Nacionalidad:

Direccion

Calle:

Transversal:

Piso:

Numero:

Urbanizacion:

Sector:

Referencia:

Ciudad:

Pais:

Sitio:

Institucion

Nombre:

Cerrar Sesión / Logout / Salir

- Presionamos la tecla TAB y los datos de la persona se despliegan automáticamente.

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales Contactos Profesores Estudiantes

Ingreso y Mantenimiento de Personas

Datos Personales

Id:

Cedula:

Nombres:

Apellidos:

Fecha de Nacimiento:

Sexo

Masculino:

Femenino:

Estado Civil:

Nacionalidad:

Direccion

Calle:

Transversal:

Piso:

Numero:

Urbanizacion:

Sector:

Referencia:

Ciudad:

Pais:

Sitio:

Institucion

Nombre:

Cerrar Sesión / Logout / Salir

- Aparecen los datos de la persona que incluyen: Nombres, Apellidos, Fecha de Nacimiento, Sexo, Estado Civil, Nacionalidad, Dirección e Institución a la que pertenece la persona.
- En la pestaña Contactos pueden verse los diferentes contactos de la persona

BIENVENIDO/A FABRICIO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales **Contactos** Profesores Estudiantes

Crear un nuevo contacto Eliminar Contactos

Sitio	Tipo Contacto	Valor	Fecha Ingreso	Fecha Modificación
CASA	TELEFONO	2694833		
MOVIL	CELULAR	087036772		
CASA	EMAIL	glorisf7@hotmail.cc		
CASA	EMAIL	glorysweets001@	2008-09-16 00:00:00	2008-09-16 00:00:00
MOVIL	CELULAR	094667074	2008-10-25 00:00:00	0000-00-00 00:00:00
CASA	TELEFONO	6036723	2008-12-07 15:14:1	0000-00-00 00:00:00

Page 1 of 1

Displaying 1 - 6 of 6

Cerrar Sesión / Logout / Salir

- Haciendo clic en la pestaña Profesores obtenemos: Datos del Empleado
- Aquí se puede ver el Empleado Id y el Profesor Id junto al estado del profesor que puede ser Activo o Inactivo.
- Seleccionamos la sección Idiomas , aparecen los idiomas que enseña el profesor en cuestión

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales Contactos Profesores Estudiantes

DATOS DE EMPLEADO

IDIOMAS

Crear un nuevo idioma Eliminar idiomas

#	Profesor Id	Estudiante Id	Idioma Id	Idioma
12	9	1		INGLES
28	9	2		FRANCES

Page 1 of 1

Displaying 1 - 2 of 2

DISPONIBILIDAD

INFORMACION SALARIAL

Cerrar Sesión / Logout / Salir

- Seleccionamos la sección disponibilidad y se despliegan las disponibilidades del profesor

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales Contactos Profesores Estudiantes

DATOS DE EMPLEADO

IDIOMAS

DISPONIBILIDAD

Crear disponibilidad Eliminar Disponibilidad

#	Profesor Id	Idioma	Dia Disponible	Fecha Ingreso	Fecha Modificacion	Comienzo	Fin
24	9	INGLES	LUNES	2008-12-11 00:01:00	2009-10-04 05:22:26	7:00 AM	1:00 PM
25	9	INGLES	MARTES	2008-12-11 00:01:18	2009-11-22 02:21:54	7:00 AM	1:00 PM
50	9	INGLES	MIERCOLES	2010-05-11 18:15:43	2010-05-15 20:37:12	7:00 AM	9:00 AM

Page 1 of 1

Displaying 1 - 3 of 3

INFORMACION SALARIAL

Cerrar Sesión / Logout / Salir

- Pulsando en la sección Información Salarial se despliegan los datos de salarios según la función que cumple el profesor

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales Contactos Profesores Estudiantes

DATOS DE EMPLEADO

IDIOMAS

DISPONIBILIDAD

INFORMACION SALARIAL

Crear salario Eliminar salario

Posicion	Tipo Posicion	Salario
PROFESOR	HORAS 2	4.50
PROFESOR	TIEMPO COMPLETO 3	900.00
SUSTITUTO	HORAS 2	3.50
ASISTENTE DE PROFESOR	TIEMPO COMPLETO 3	400.00
ASISTENTE DE PROFESOR	HORAS 2	3.00

Page 1 of 1

Displaying 1 - 5 of 5

Cerrar Sesión / Logout / Salir

- Al hacer clic en la pestaña Estudiantes se muestra las matriculas que están activas

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales Contactos Profesores Estudiantes

MATRICULAS ACTUALES

#	Idioma	Modalidad	Tipo	NivelNumero	NumHoras	Fecha	Subtotal	Total	FechaModificacion	Estado
11	INGLES	INDIVIDUAL	NORMAL	2	20.00	2009-02-21 23:36:32	200.00	200.00	2009-04-12 22:39:16	22
12	INGLES	INDIVIDUAL	NORMAL	1	20.00	2009-04-05 13:44:44	215.00	215.00	2009-04-05 13:45:41	22
13	FRANCES	INDIVIDUAL	NORMAL	1	0.00	2009-06-07 12:37:19	350.00	350.00	0000-00-00 00:00:00	22
14	ITALIANO	INDIVIDUAL	NORMAL	1	0.00	2009-06-24 21:26:36	350.00	350.00	2009-06-25 03:38:34	22
56	INGLES	INDIVIDUAL	NORMAL	1	2.50	2009-11-07 10:24:48	25.00	25.00	0000-00-00 00:00:00	22

Page 1 of 1

Displaying 1 - 5 of 5

MATRICULAS HISTORIAL

Cerrar Sesión / Logout / Salir

Resultado esperado

El sistema nos muestra los datos personales , datos de contactos , datos de empleado y estudiante.

Evaluación de la Prueba

Prueba superada con éxito

Prueba 2: Comprobar que el sistema almacena los datos de una persona correctamente.

Descripción

Ingresamos al sistema como Operador, accediendo a la funcionalidad de Ingreso y Mantenimiento de Datos.

Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el sistema busque una cedula existente y al no encontrar dicha cedula permita el ingreso de todos los datos de la persona.

Entrada

- Introducimos la cedula “0909158313” en el campo Cedula

Lincoln College of Foreign Languages Proceso de Matriculación

Bienvenido/a FABRICIO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales Contactos Profesores Estudiantes

Ingreso y Mantenimiento de Personas

Datos Personales	Direccion
Id: <input type="text"/>	Calle: <input type="text"/>
Cedula: <input type="text" value="0909158313"/>	Transversal: <input type="text"/>
Nombres: <input type="text" value="Ingrese los nombres"/>	Piso: <input type="text"/>
Apellidos: <input type="text" value="Ingrese los apellidos"/>	Numero: <input type="text"/>
Fecha de Nacimiento: <input type="text"/>	Urbanizacion: <input type="text"/>
Sexo	Sector: <input type="text"/>
Masculino: <input type="radio"/>	Referencia: <input type="text"/>
Femenino: <input type="radio"/>	Ciudad: <input type="text"/>
Estado Civil: <input type="text"/>	Pais: <input type="text"/>
Nacionalidad: <input type="text"/>	Sitio: <input type="text"/>
	Institucion
	Nombre: <input type="text"/>

Cerrar Sesion / Logout / Salir

- No aparecen datos de la persona , así que vamos a ingresar a esta persona en el sistema.
- Ingresamos los datos personales

BIENVENIDO/a FABRICIO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales Contactos Profesores Estudiantes

Ingreso y Mantenimiento de Personas

Datos Personales

Id:

Cedula: 0909158313

Nombres: JORGE RAMIRO

Apellidos: CORTEZ MEDRANO

Fecha de Nacimiento: 11/12/1963

Sexo

Masculino:

Femenino:

Estado Civil: SOLTERO

Nacionalidad: ECUATORIANA

Direccion

Calle: LADRON DE GUEVARA

Transversal: AV. 12 DE OCTUBRE

Piso: 2

Numero: 344

Urbanizacion: LOS CEIBOS

Sector: LA UNIVERSIDAD

Referencia: FRENTE A IGLESIA

Ciudad: QUITO

Pais: ECUADOR

Sitio: TRABAJO

Institucion

Nombre: Municipio de Quito

Guardar Cancelar

Cerrar Sesión / Logout / Salir

- Presionamos el botón Guardar .

BIENVENIDO/a FABRICIO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales Contactos Profesores Estudiantes

Ingreso y Mantenimiento de Personas

Datos Personales

Id: 50

Cedula: 0909158313

Nombres: JORGE RAMIRO

Apellidos: CORTEZ MEDRANO

Fecha de Nacimiento: 11/12/1963

Sexo

Masculino:

Femenino:

Estado Civil: SOLTERO

Nacionalidad: ECUATORIANA

Direccion

Calle: LADRON DE GUEVARA

Transversal: AV. 12 DE OCTUBRE

Piso: 2

Numero: 344

Urbanizacion: LOS CEIBOS

Sector: LA UNIVERSIDAD

Referencia: FRENTE A IGLESIA

Ciudad: QUITO

Pais: ECUADOR

Sitio: TRABAJO

Institucion

Nombre: Municipio de Quito

Guardar Cancelar

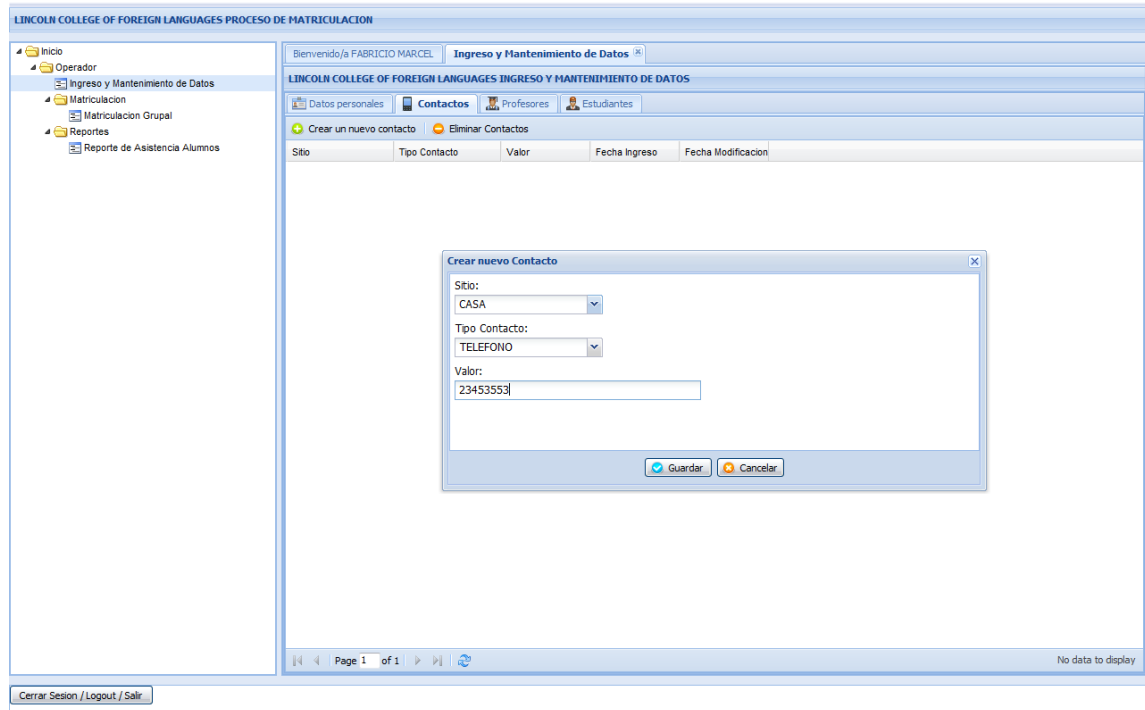
Creación Exitosa

La persona ha sido creada en el sistema.

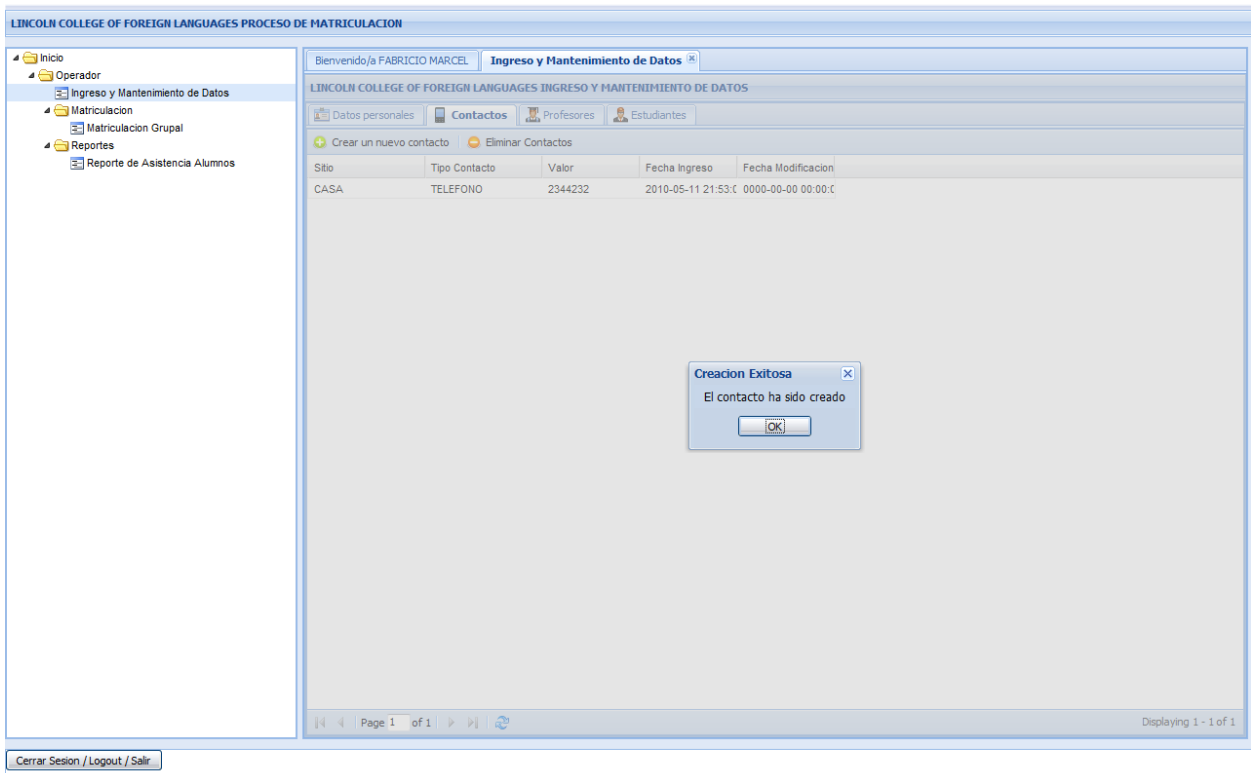
OK

Cerrar Sesión / Logout / Salir

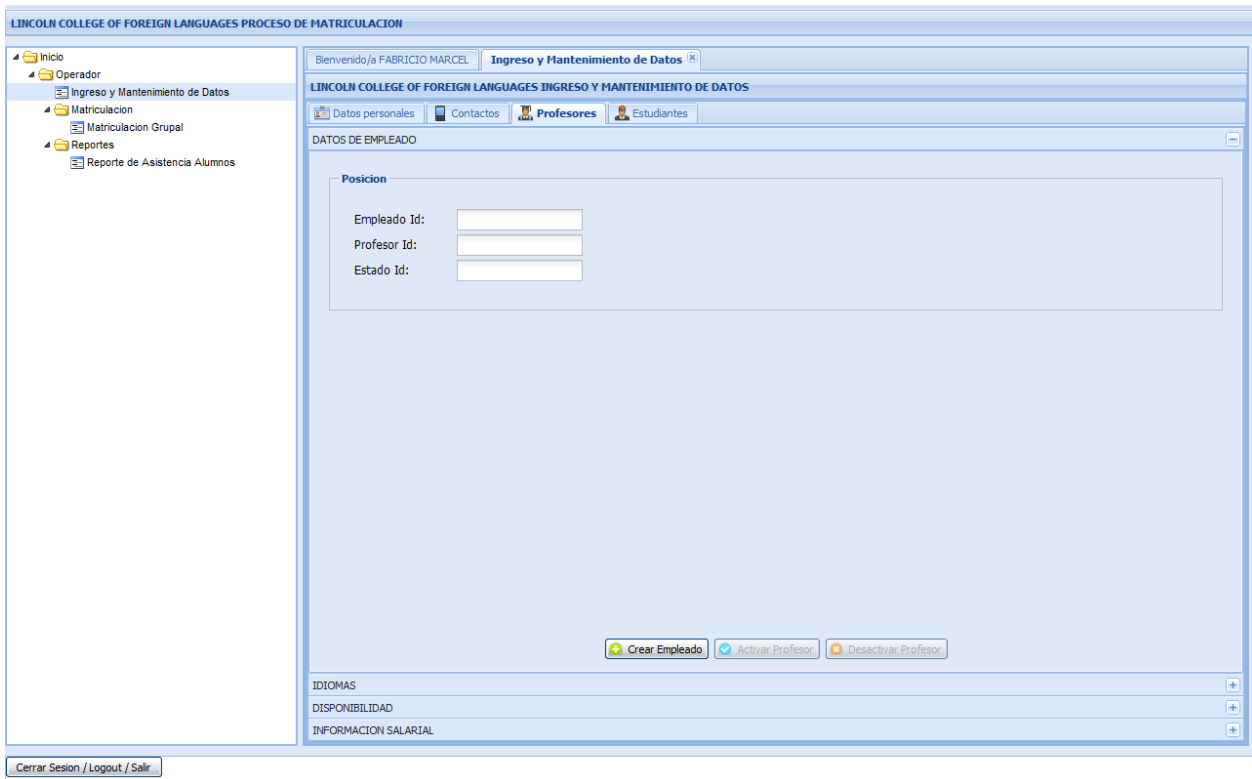
- Ahora ingresamos los contactos . Ingresamos el Sitio , Tipo de Contacto y el valor.



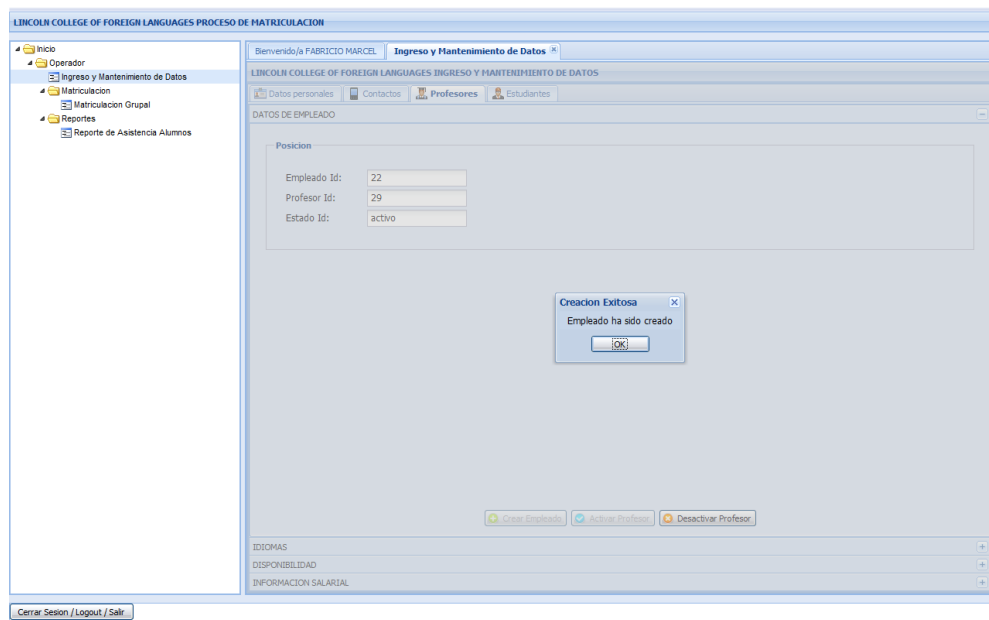
- Al presionar el botón Guardar aparece el registro en la tabla más un mensaje de confirmación.



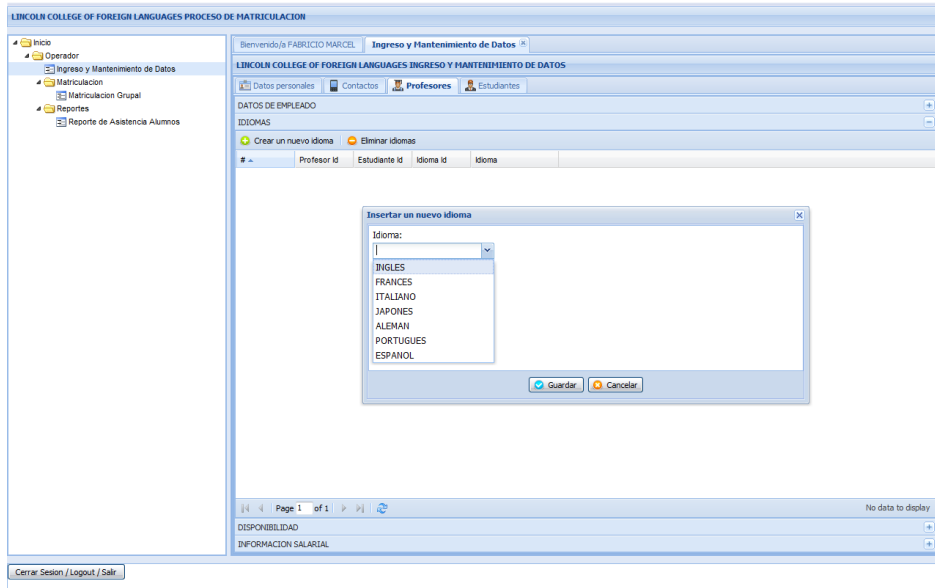
- Ahora vamos a hacer que esta persona sea un profesor por que hacemos clic en la pestaña Profesores. Para crear un empleado presionamos el botón Crear Empleado



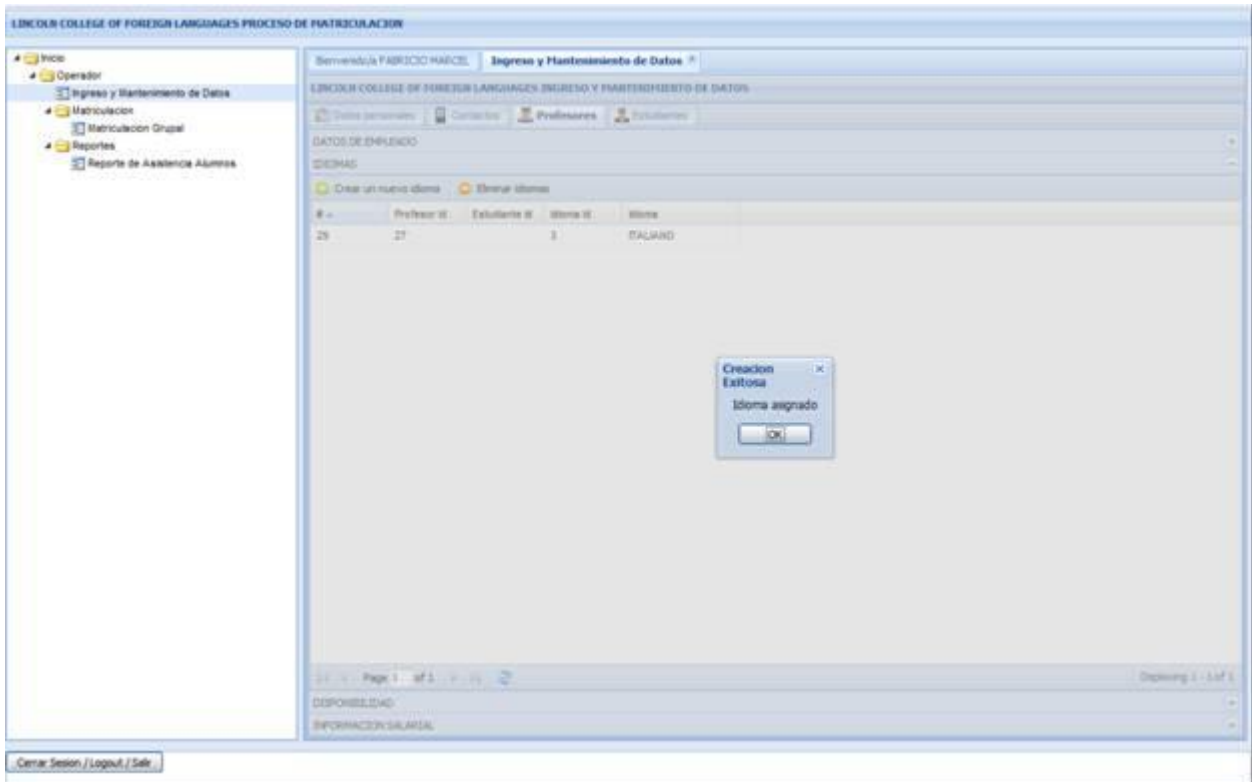
- Se crea entonces una identificación de empleado y de profesor para la cual se despliega un mensaje de confirmación.



- Seleccionamos la sección Idiomas y creamos los idiomas del profesor.



- Al seleccionar un idioma de la lista desplegable y presionar el botón Guardar se despliega la ventana de diálogo de confirmación.



- El idioma asignado aparece en la tabla.

Lincoln College of Foreign Languages Proceso de Matriculación

Bienvenido/a FABRIZO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales Contactos Profesores Estudiantes

DATOS DE EMPLEADO

IDIGOMAS

Crear un nuevo idioma Eliminar idiomas

#	Profesor id	Estudiante id	Idioma
29	27	3	ITALIANO

Page 1 of 1

Displaying 1 - 1 of 1

DISPONIBILIDAD

INFORMACION SALARIAL

Cerrar Sesión / Logout / Salir

- Pulsamos la sección Disponibilidad e ingresamos-seleccionamos los datos de Idioma, Día , hora inicio , hora fin.

Lincoln College of Foreign Languages Proceso de Matriculación

Bienvenido/a FABRIZO MARCEL Ingreso y Mantenimiento de Datos

LINCOLN COLLEGE OF FOREIGN LANGUAGES INGRESO Y MANTENIMIENTO DE DATOS

Datos personales Contactos Profesores Estudiantes

DATOS DE EMPLEADO

IDIGOMAS

DISPONIBILIDAD

Crear disponibilidad Eliminar Disponibilidad

Profesor id

Idioma: ITALIANO

Día Disponible: LUNES

Hora Inicio: 7:00 AM

Hora Fin: 8:00 AM

Guardar Cancelar

Page 1 of 1

No data to display

INFORMACION SALARIAL

Cerrar Sesión / Logout / Salir

- Si la disponibilidad ha sido creada se despliega el siguiente mensaje.

The screenshot shows the 'INGRESO Y MANTENIMIENTO DE DATOS' section of the application. A modal dialog box titled 'Creacion Exitosa' is displayed in the center, with the message 'La disponibilidad ha sido creada' and an 'OK' button. The background interface shows a table of availability records.

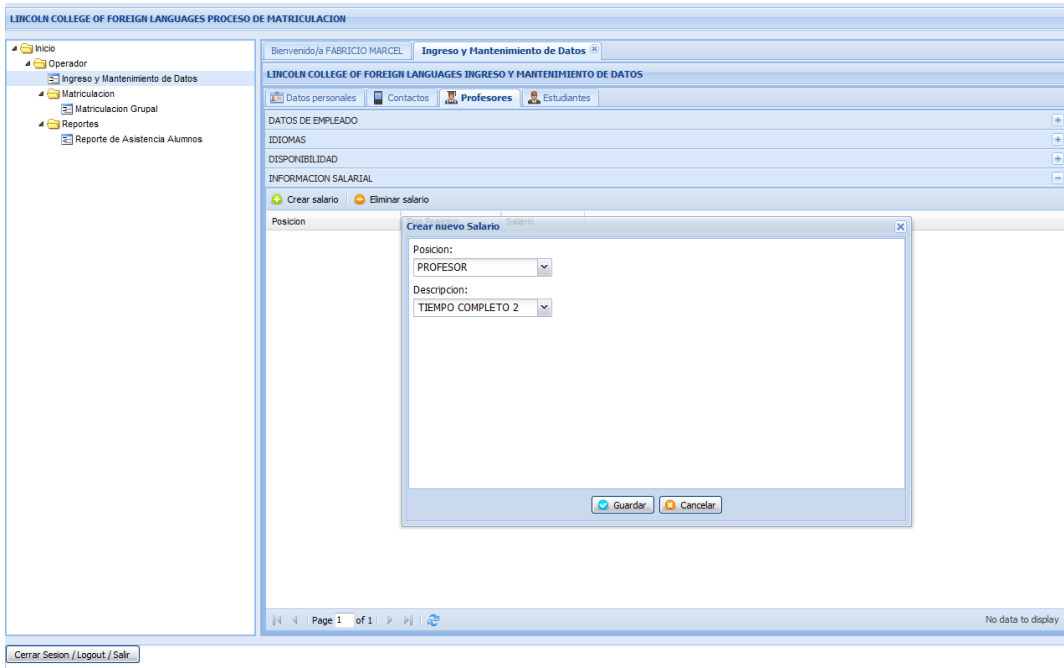
#	Profesor Id	Idioma	Dia Disponible	Fecha Ingreso	Fecha Modificacion	Comienzo	Fin
51	27	ITALIANO	LUNES	2010-05-11 22:09:25	0000-00-00 00:00:00	7:00 AM	8:00 AM
52	27	ITALIANO	JUEVES	2010-05-11 22:11:39	0000-00-00 00:00:00	7:00 AM	8:00 AM

- Ahora ingresemos una configuración salarial. Primero escoger el área Información Salarial.

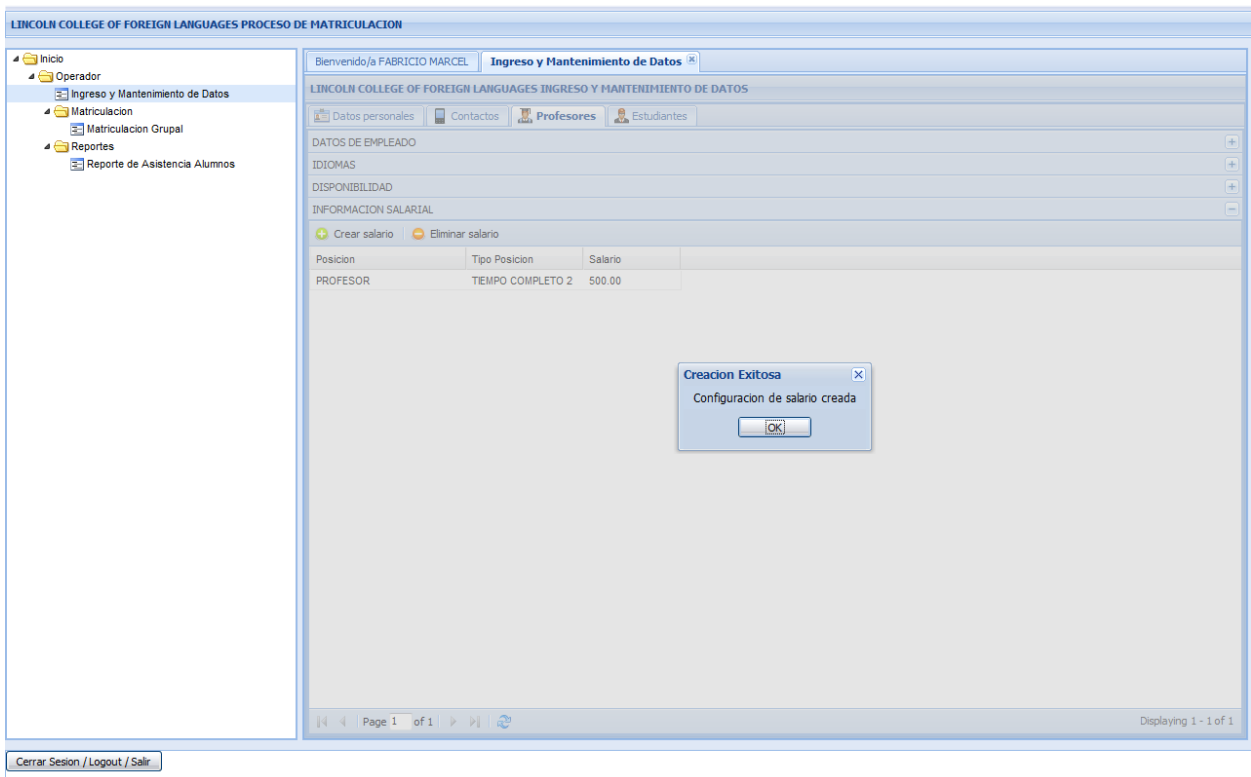
The screenshot shows the 'INFORMACION SALARIAL' section of the application. The interface includes a navigation menu on the left and a main content area with a table for salary configuration.

Posicion	Tipo Posicion	Salario
No data to display		

- Presionamos el botón Crear Salario para ingresa nuevas configuraciones, Seleccionamos la posición, y la descripción de la configuración salarial.



Al presionar el botón Guardar , aparece la ventana de confirmación de creación exitosa.



- Al presionar el botón Ok , la disponibilidad aparece en la tabla.

Resultado esperado

El sistema guarda correctamente los datos de ingreso y mantenimiento.

Evaluación de la Prueba

Prueba superada con éxito.

FASE: CONSTRUCCIÓN	ID: CNS002
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de Prueba: Matricular Alumnos
VERSIÓN:	1.0

Descripción

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Matricular Alumnos”.

Se ingresa o actualiza datos con el perfil de Operador. Este caso de uso incluye a los casos de uso Crear Curso y Asignar Profesor.

Comprobar que el sistema matricula uno o más alumnos en un curso de idiomas.

Descripción

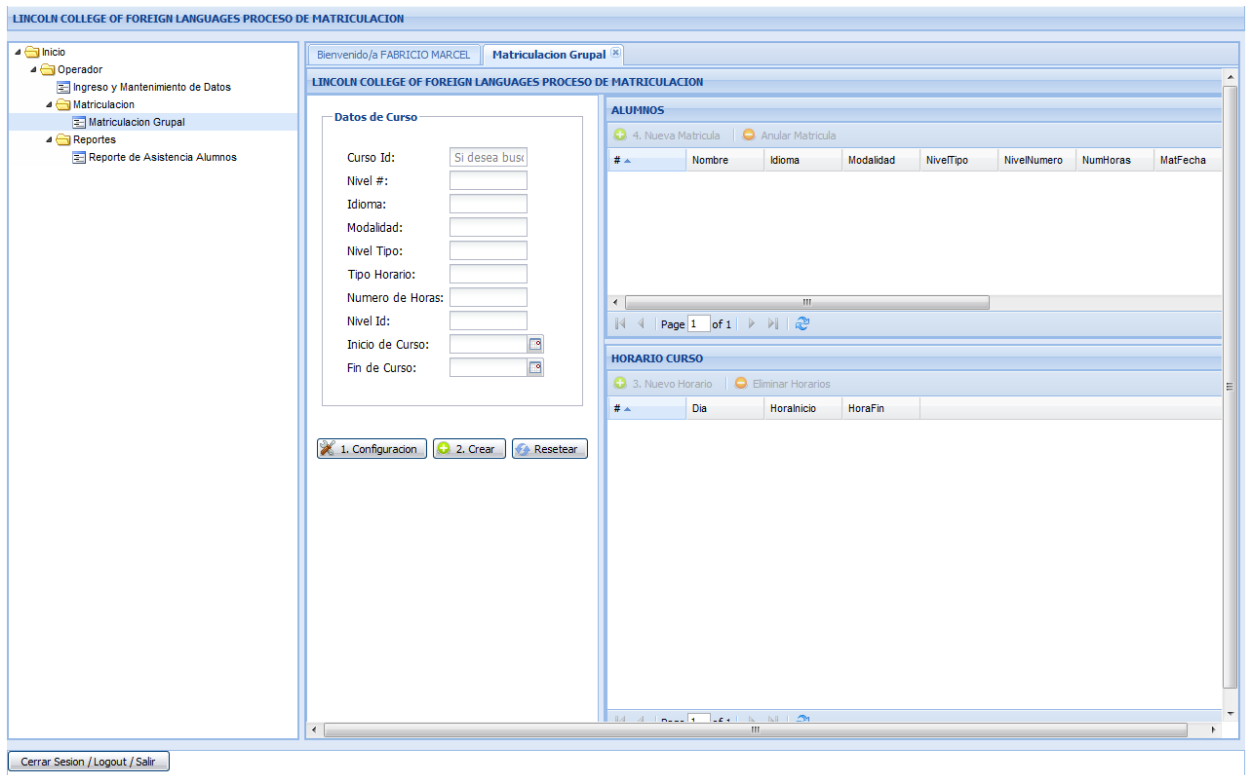
Ingresamos al sistema como Operador , este perfil ofrece la opción de ejecutar el proceso de matriculación.

Condiciones de ejecución

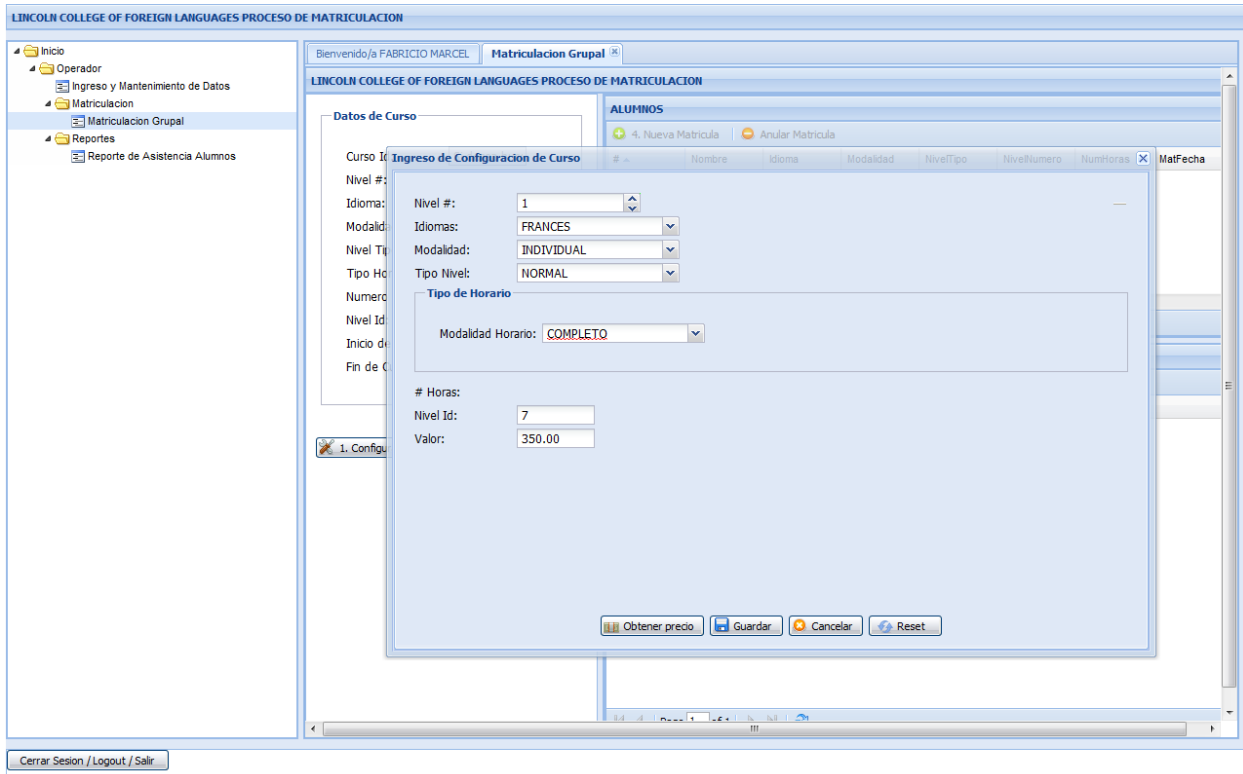
Las condiciones de ejecución del caso de prueba son que el sistema permita matricular un alumno en un curso de idiomas.

Entrada

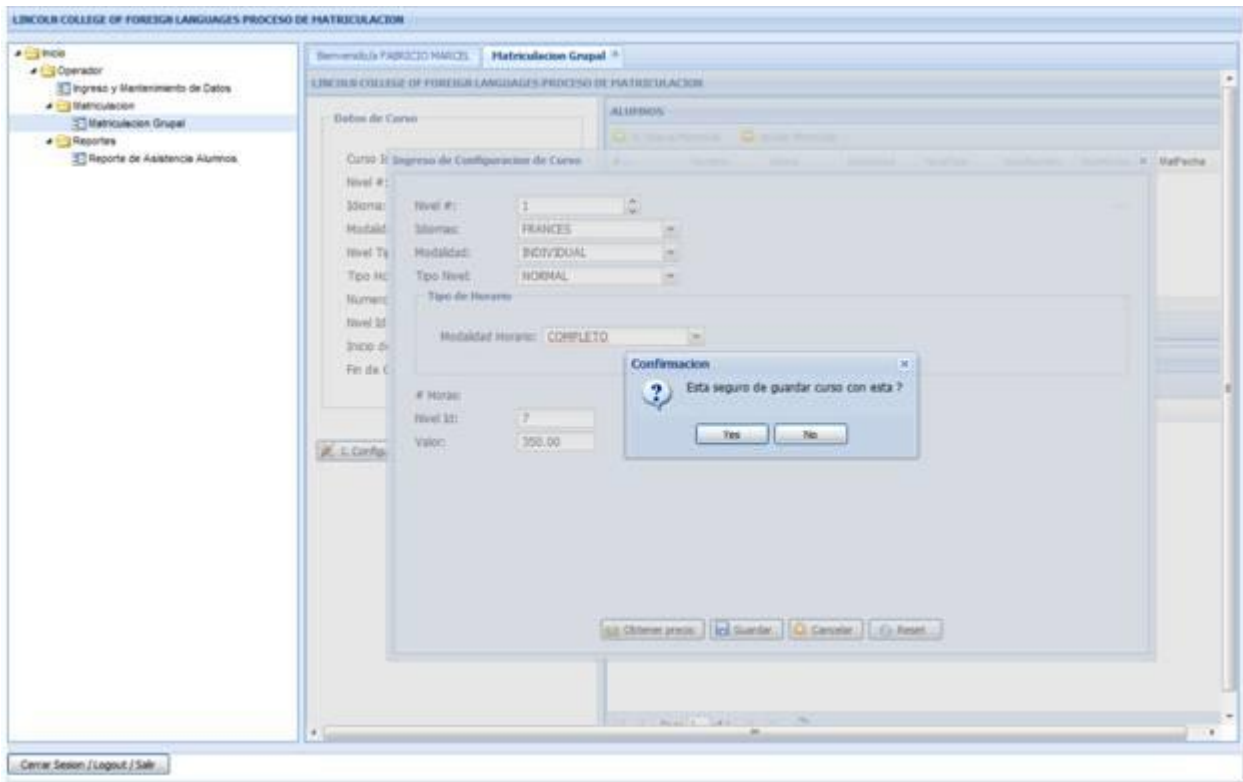
- Seleccionamos la opción “Matriculación Grupal” del Viewport , lo cual despliega la siguiente pantalla



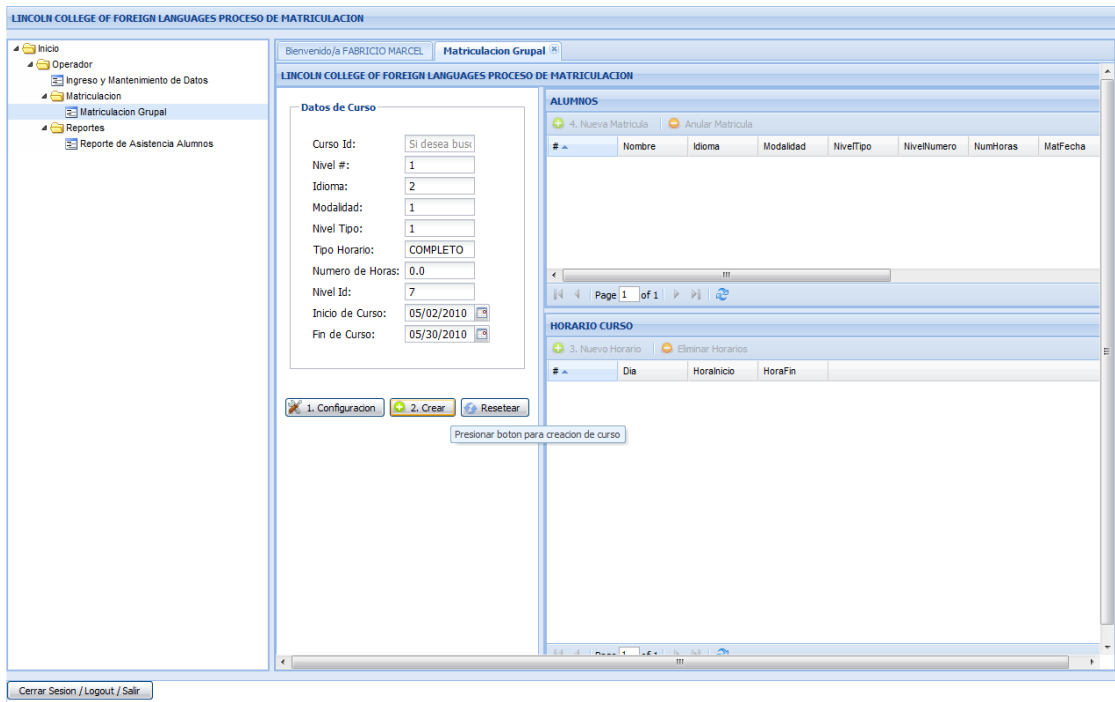
- Presionamos el botón “1.Configuracion”, para seleccionar e ingresar los parámetros de configuración del curso a crearse (Caso de Uso: Crear Curso)



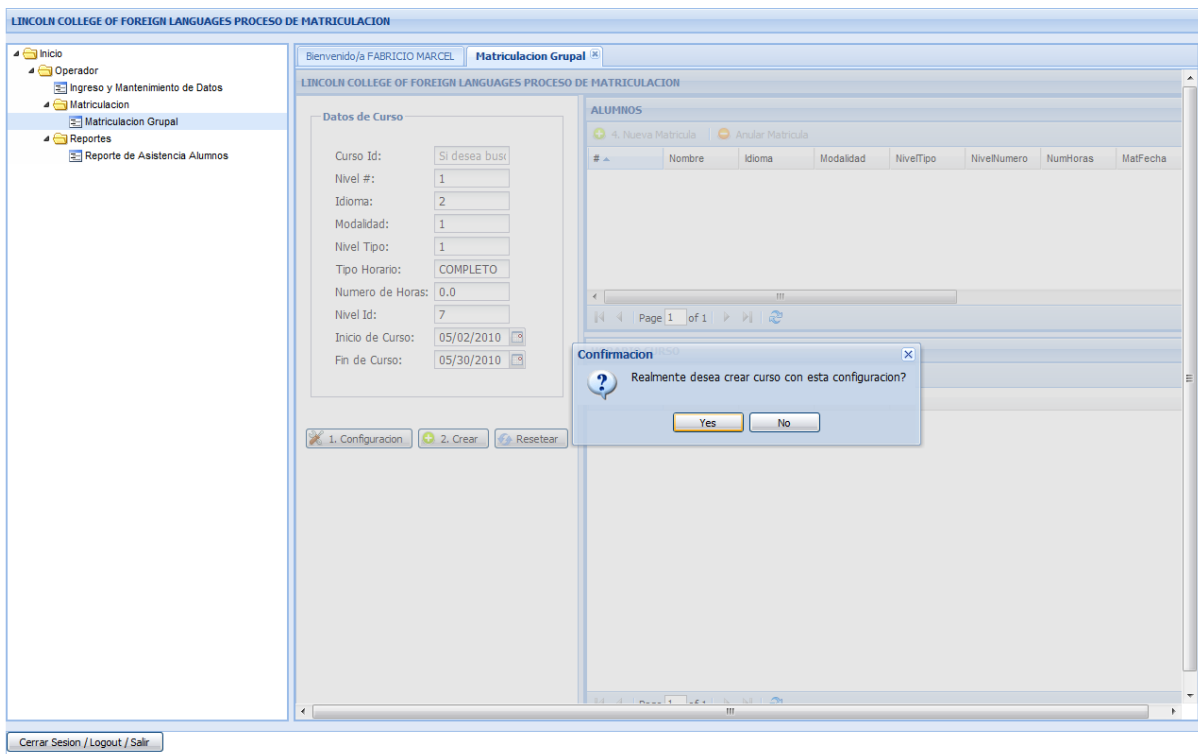
- Una vez seleccionados los parámetros de configuración presionamos el botón Guardar que nos despliega la ventana de confirmación.



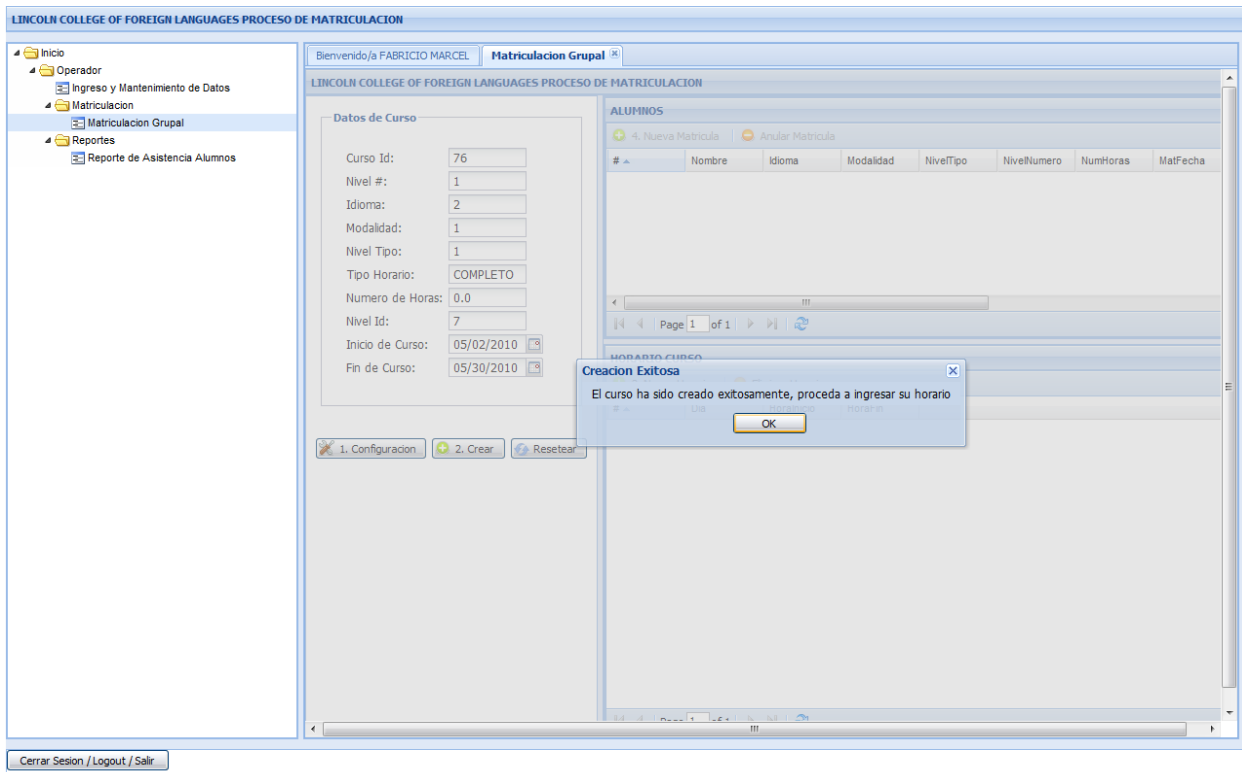
- Al presionar Si , se puede ver los datos de la configuración . Procedemos a seleccionar las fechas de inicio y fin del curso.



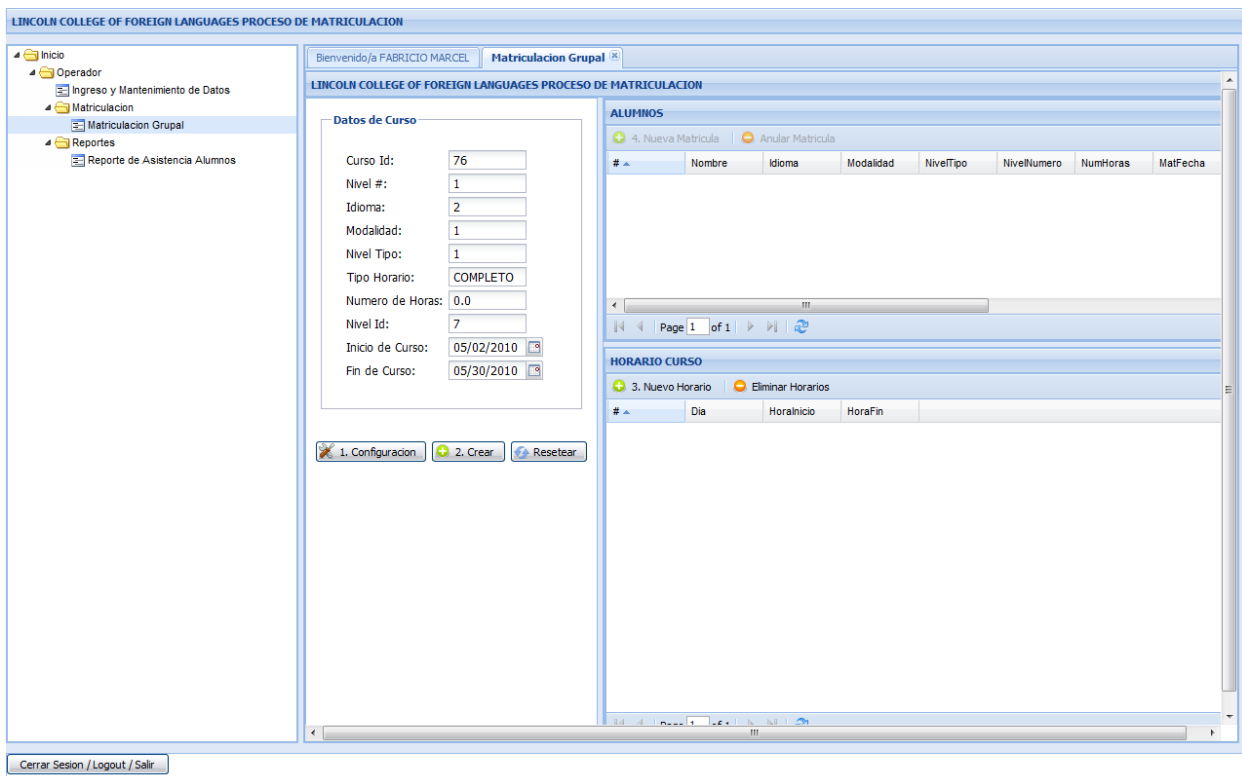
- Presionamos el botón “2.Crear”. para desplegar la ventana de confirmación.



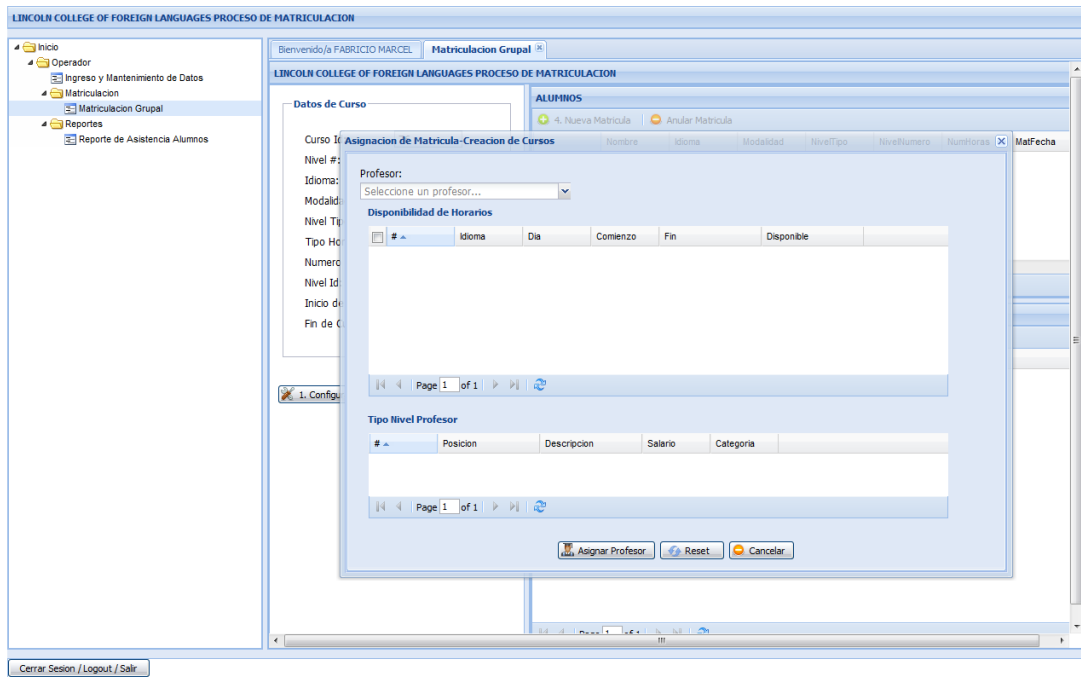
- Aceptamos la creación presionando el botón Si, para desplegar la ventana de confirmación de creación de curso.



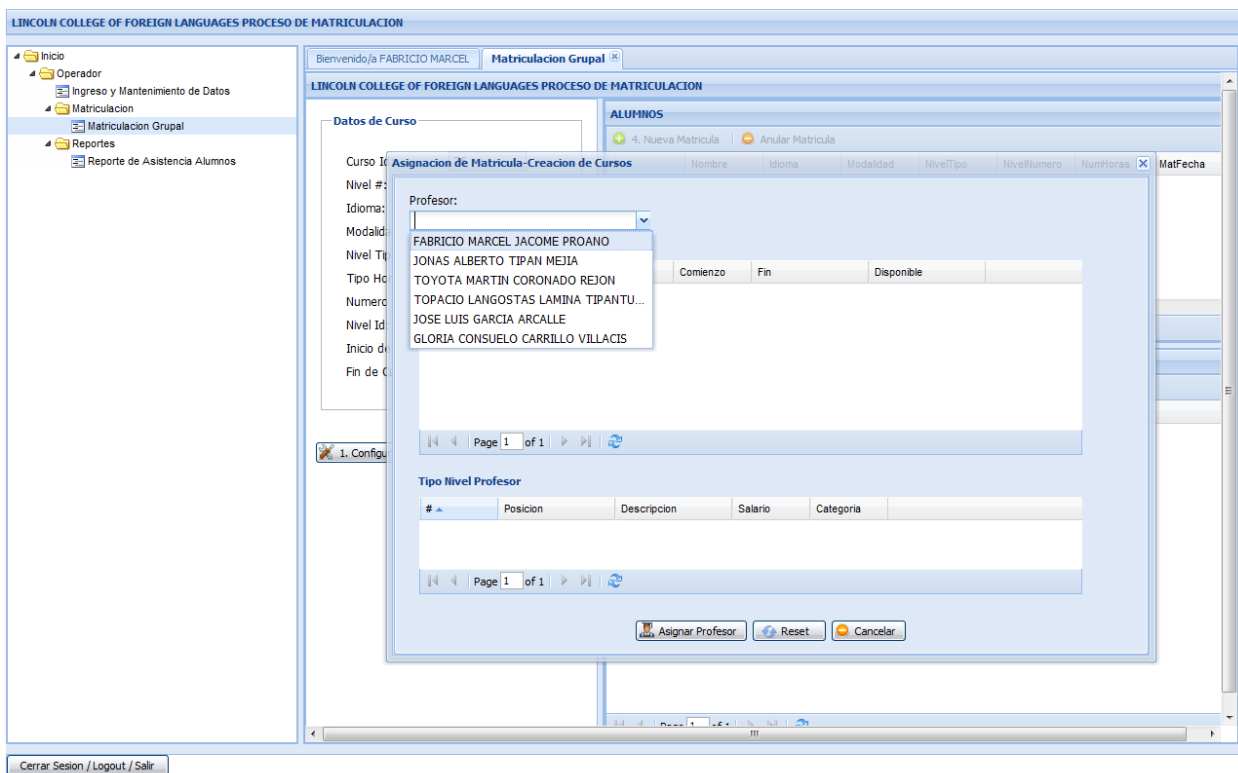
- Al presionar el botón OK de la ventana de confirmación se puede observar el campo Curso Id tiene un numero asignado , en este caso el número 76. Además el botón “3. Nuevo Horario” se encuentra habilitado.



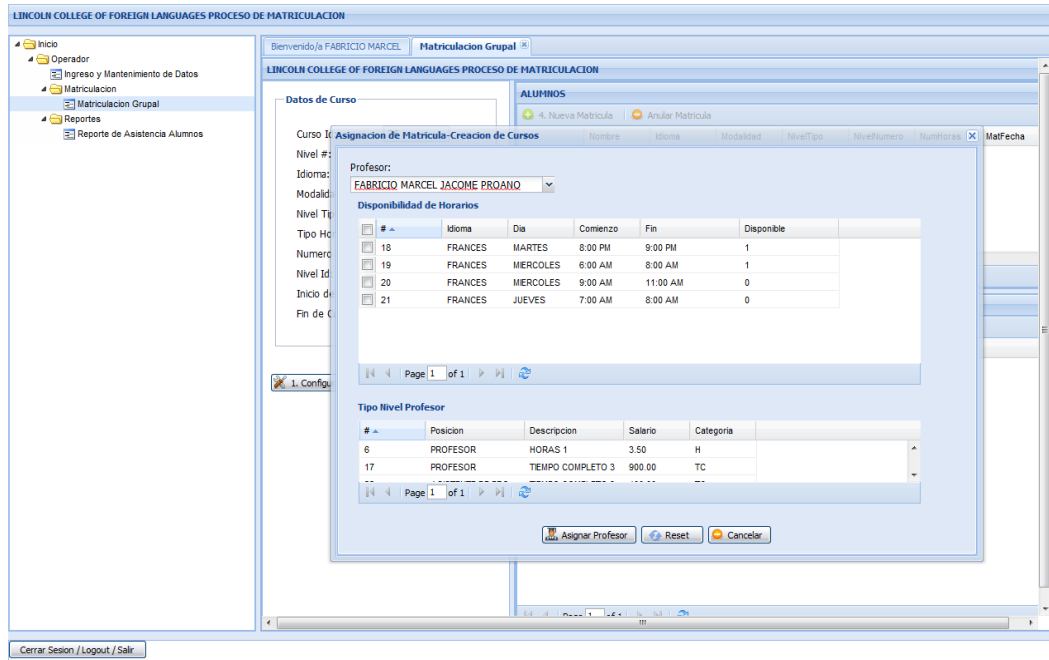
- Presionamos el botón “3. Nuevo Horario”, en donde se nos despliega una ventana.(Caso de Uso: Asignar Profesor)



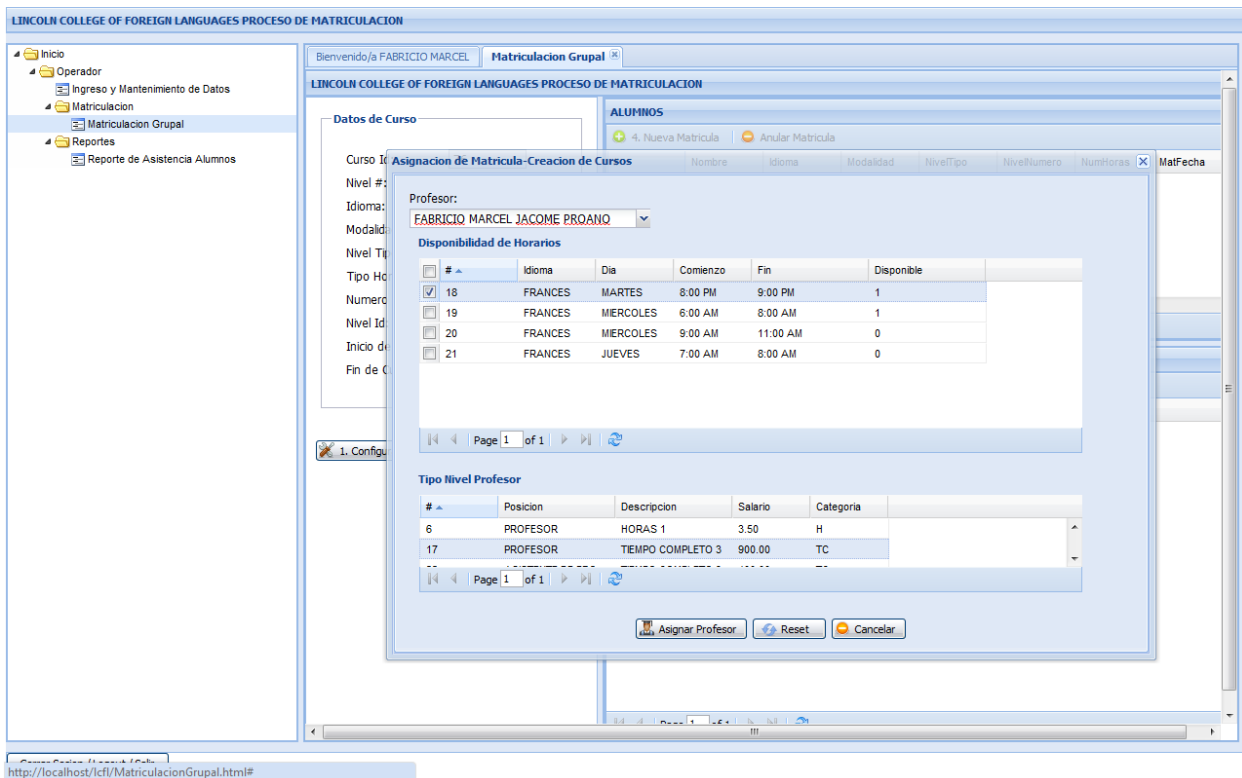
- Presionamos el combo Profesor para seleccionar un profesor para el curso recién creado. Se nos despliega una lista de los profesores que enseñan el idioma FRANCES en este caso.



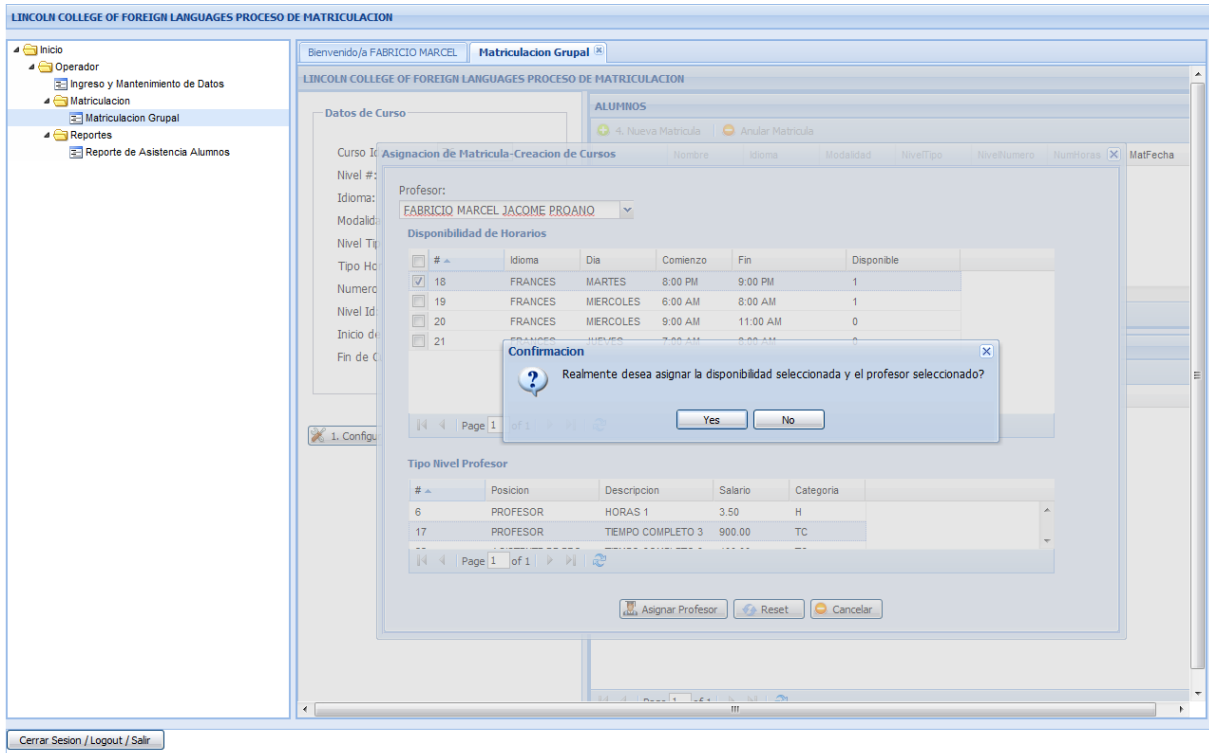
- Seleccionamos el profesor “FABRICIO MARCEL JACOME PROAÑO” y se nos despliega las disponibilidades además de las configuraciones salariales



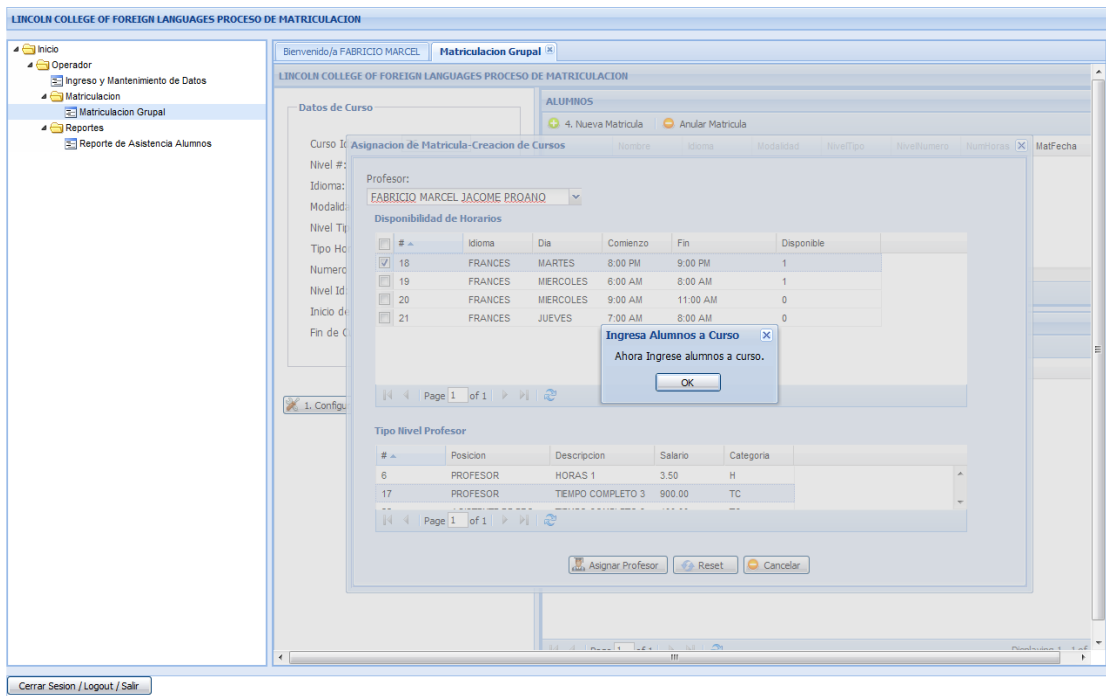
- Seleccionamos la configuración con Id: 18 la cual está disponible, como se puede ver en el campo Disponible consta un 1. También seleccionamos una configuración de salario, en este caso, “PROFESOR TIEMPO COMPLETO 3”



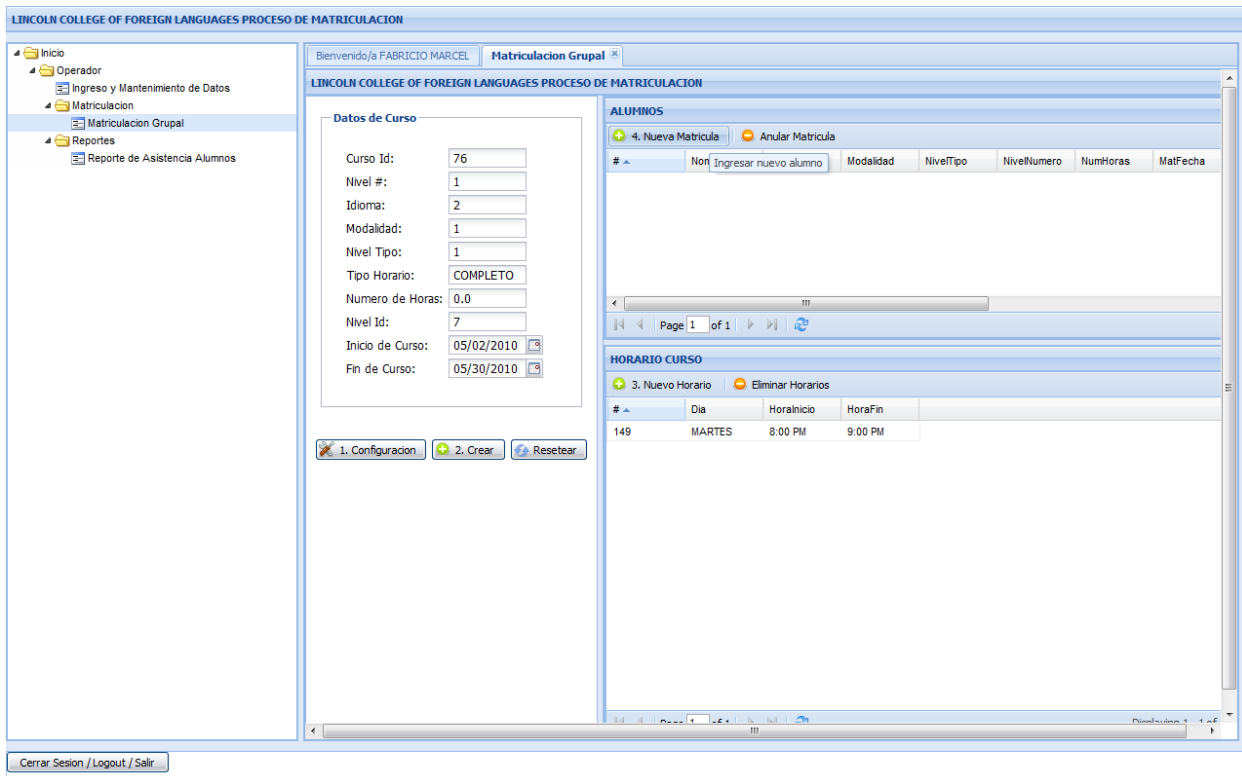
- Presionamos el botón Asignar Profesor y aceptamos la confirmación.



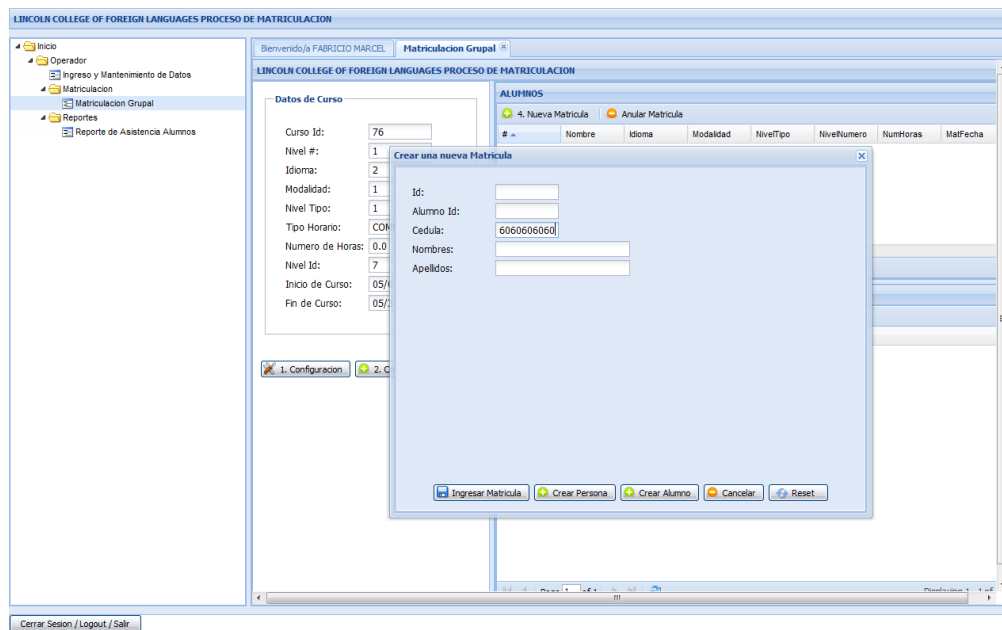
- Presionamos el botón Asignar Profesor y aceptamos la confirmación. Aparece el mensaje “Ahora Ingresar Alumnos a Curso”



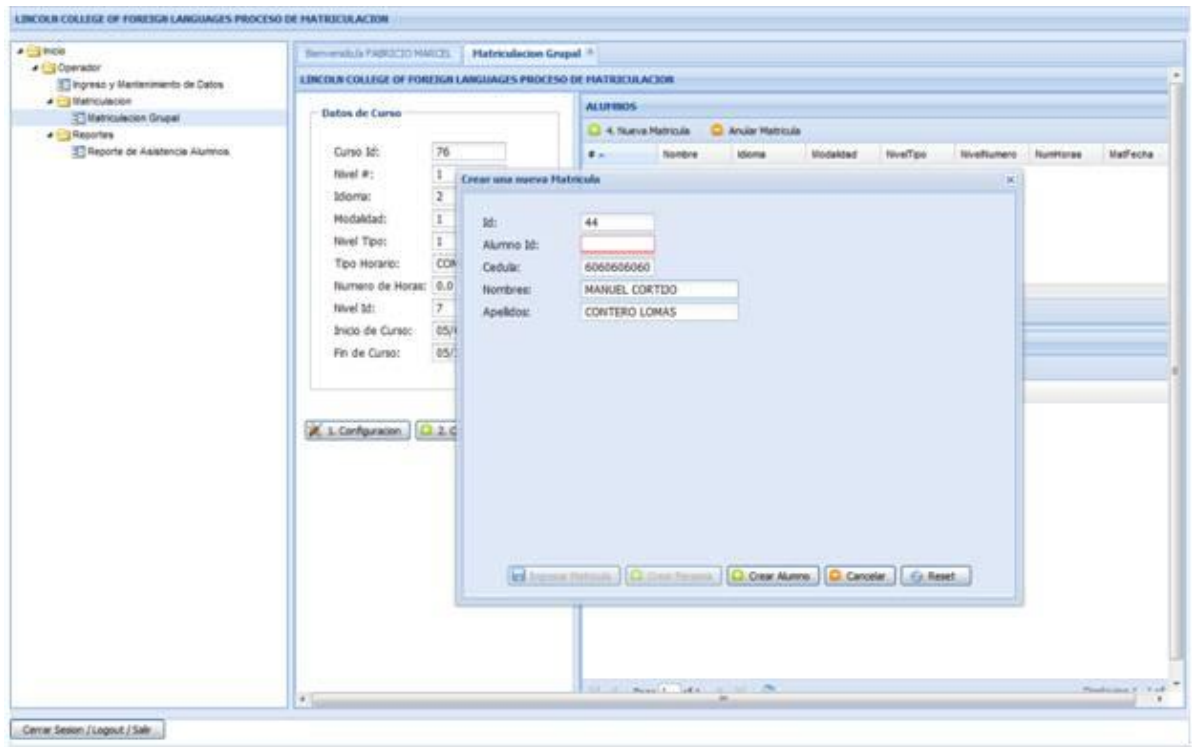
- Presionamos el botón Asignar Profesor y aceptamos . Aparece el mensaje “Ahora Ingresar Alumnos a Curso”. Podemos ver que el horario aparece en la sección HORARIO CURSO y el botón “4.Nueva Matricula” se encuentra habilitado.



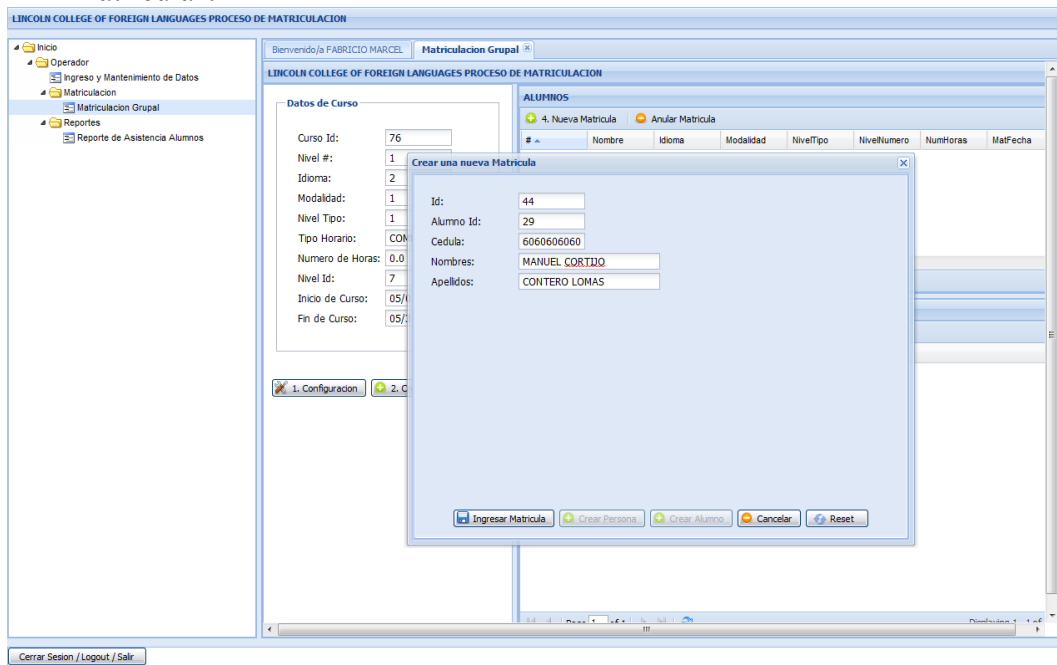
- Presionamos el botón “4. Nueva Matricula” donde ingresamos la cedula del alumno que se va a matricular, en este caso “6060606060”.



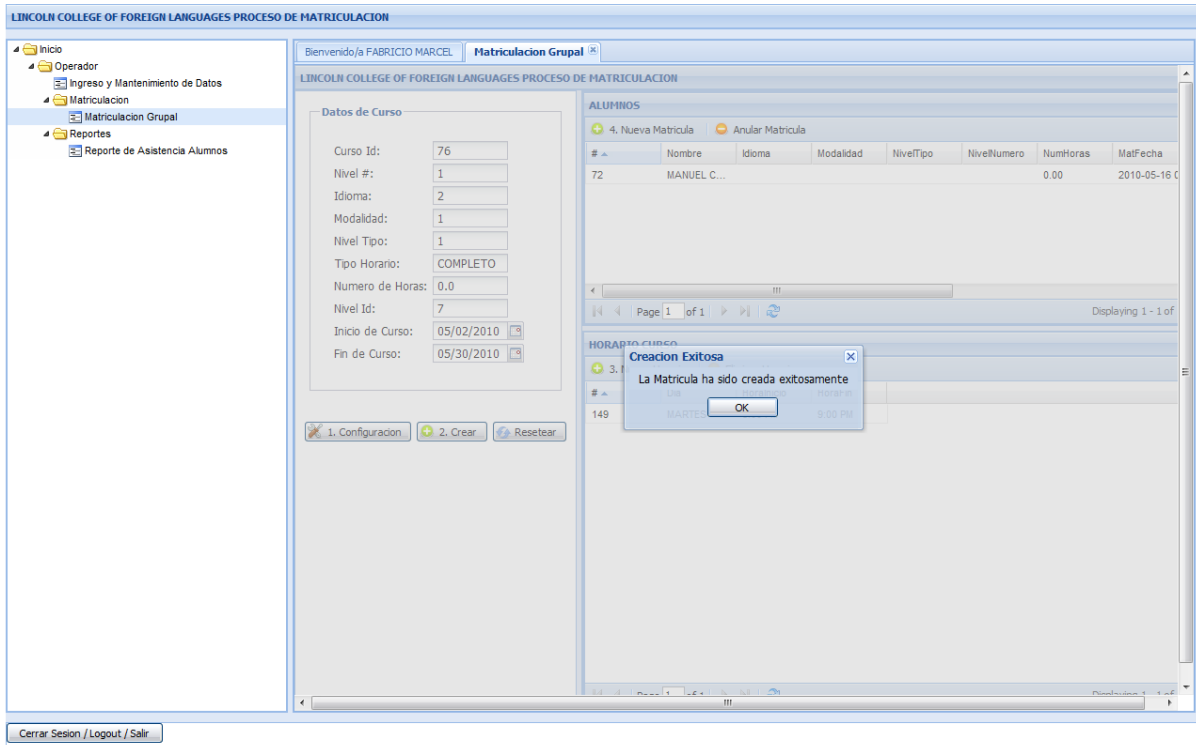
- Una vez retirado el enfoque del campo cédula, se despliega los nombres y apellidos además del Id de la persona , en este caso 44. Permanecen deshabilitados los botones “Ingresar Matricula” y “Crear Persona”



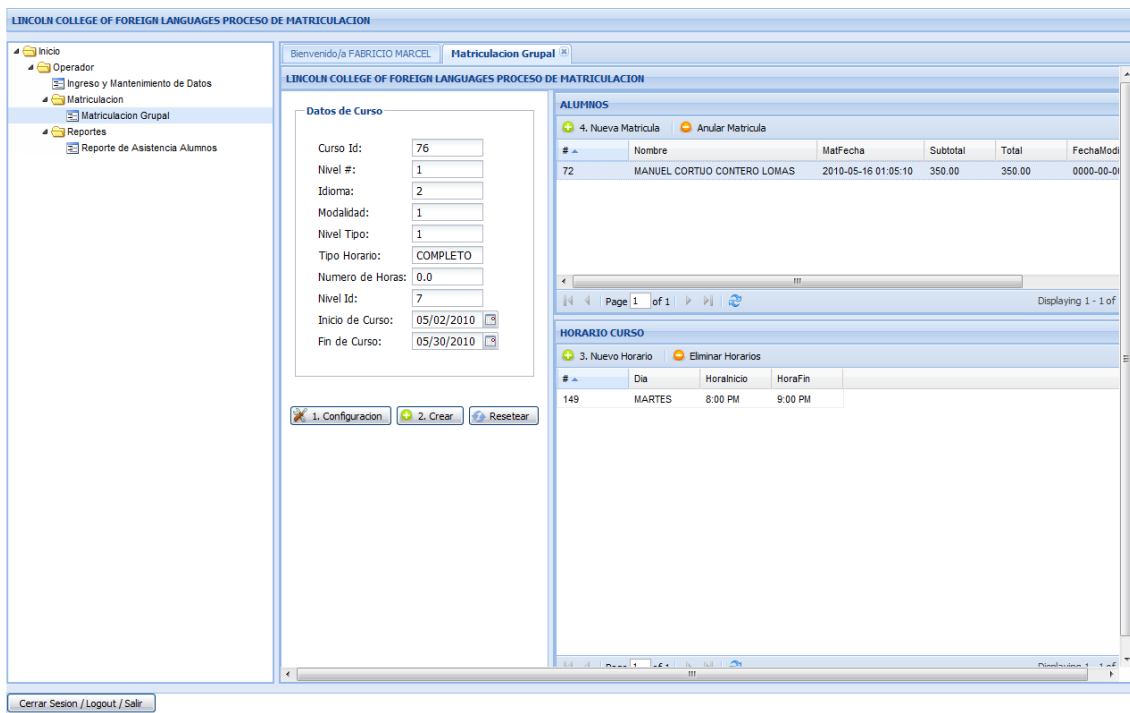
- Presionamos el botón “Crear Alumno”. y aparece un número de alumno en Alumno Id. Se deshabilita el botón “Crear Alumno” y se habilita el botón “Ingresar Matricula”.



- Presionamos el botón “Ingresar Matricula”. y aparece una ventana de confirmación



- Al presionar OK , se puede ver el registro de matricula creado.



- Repetimos la operación de matriculación con la cedula “1709771438” para matricular un alumno más.

Lincoln College of Foreign Languages Proceso de Matriculación

Bienvenido/a FABRICIO MARCEL | Matriculación Grupal

Inicio

- Operador
- Ingreso y Mantenimiento de Datos
- Matriculación
 - Matriculación Grupal
 - Reportes
 - Reporte de Asistencia Alumnos

Datos de Curso

Curso Id: 76
 Nivel #: 1
 Idioma: 2
 Modalidad: 1
 Nivel Tipo: 1
 Tipo Horario: COMPLETO
 Numero de Horas: 0.0
 Nivel Id: 7
 Inicio de Curso: 05/02/2010
 Fin de Curso: 05/30/2010

1. Configuración | 2. Crear | Resetear

ALUMNOS

4. Nueva Matricula | Anular Matricula

#	Nombre	MatFecha	Subtotal	Total	FechaModific
72	MANUEL CORTUO CONTERO LOMAS	2010-05-16 01:05:10	350.00	350.00	0000-00-00 0
73	CARLOS JACOME	2010-05-16 01:09:53	350.00	350.00	0000-00-00 0

Page 1 of 1 | Displaying 1 - 2 of 2

HORARIO CURSO

5. Nuevo Horario | Eliminar Horarios

#	Dia	Horainicio	HoraFin
149	MARTES	8:00 PM	9:00 PM

Page 1 of 1 | Displaying 1 - 1 of 1

http://localhost/ctf/MatriculacionGrupal.html#

Resultado esperado

El sistema crea un curso, asigna un profesor y matricula uno o más alumnos.

Evaluación de la Prueba

Prueba superada con éxito.

FASE: CONSTRUCCIÓN	ID: CNS003
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCION:	Especificación de Caso de Prueba: Mantener Parámetros Corporativos
VERSIÓN:	1.0

Descripción

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Mantener Parámetros Corporativos”.
Se ingresa o actualiza datos con el perfil de Administrador.

Prueba 1: Comprobar que el sistema crea un parámetro corporativo.

Descripción

Ingresamos al sistema como Administrador , este perfil ofrece la opción de ingresar y cambiar parámetros corporativos .

Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el sistema permita la creación de uno o más parámetros corporativos y que estos se puedan mantener.

Entrada

- Seleccionamos la opción “Sitios” del Viewport y aparece una tabla con los sitios disponibles para contactos del sistema.

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRIZO MARCEL Sitio

Inicio

- Administrador
 - Contactos
 - Sitios
 - Tipos de contactos
 - Cursos
 - Tipos de Calificaciones
 - temas
 - Modalidades Cursos
 - Niveles
 - Parametros Corporativos
 - Dias Laborables
 - Estados Civiles
 - Instituciones
 - Nacionalidades
 - Usuarios
 - Roles
 - Ingreso y Mantenimiento
 - Configuracion de Menus
 - Empleados
 - Posiciones
 - Datos de Posiciones
 - Salarios
 - Configuracion de Salarios
 - Reportes
 - Asistencia de Estudiantes
 - Rol de Pagos

Sitios

Crear Sitio Eliminar Sitio Refresh

id	Descripcion
1	CASA
2	TRABAJO
3	ESCUELA
4	UNIVERSIDAD
5	CASA CAMPO
6	AVION PRIVADO
9	MOVEL
10	NEGOCIO
11	EXTERIOR

Page 1 of 1

Mostrando registros 1 - 9 of 9

Cerrar Sesión / Logout / Salir

Find: reset Previous Highlight all Match case

- Presionamos el botón Crear Sitio e ingresamos “SITIO DE PRUEBA” en el campo Descripción.

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRIZO MARCEL Sitio

Inicio

- Administrador
 - Contactos
 - Sitios
 - Tipos de contactos
 - Cursos
 - Tipos de Calificaciones
 - temas
 - Modalidades Cursos
 - Niveles
 - Parametros Corporativos
 - Dias Laborables
 - Estados Civiles
 - Instituciones
 - Nacionalidades
 - Usuarios
 - Roles
 - Ingreso y Mantenimiento
 - Configuracion de Menus
 - Empleados
 - Posiciones
 - Datos de Posiciones
 - Salarios
 - Configuracion de Salarios
 - Reportes
 - Asistencia de Estudiantes
 - Rol de Pagos

Sitios

Crear Sitio Eliminar Sitio Refresh

id	Descripcion
1	SITIO DE PRUEBA
2	CASA
3	TRABAJO
4	ESCUELA
5	UNIVERSIDAD
6	CASA CAMPO
9	AVION PRIVADO
10	MOVEL
11	NEGOCIO
12	EXTERIOR

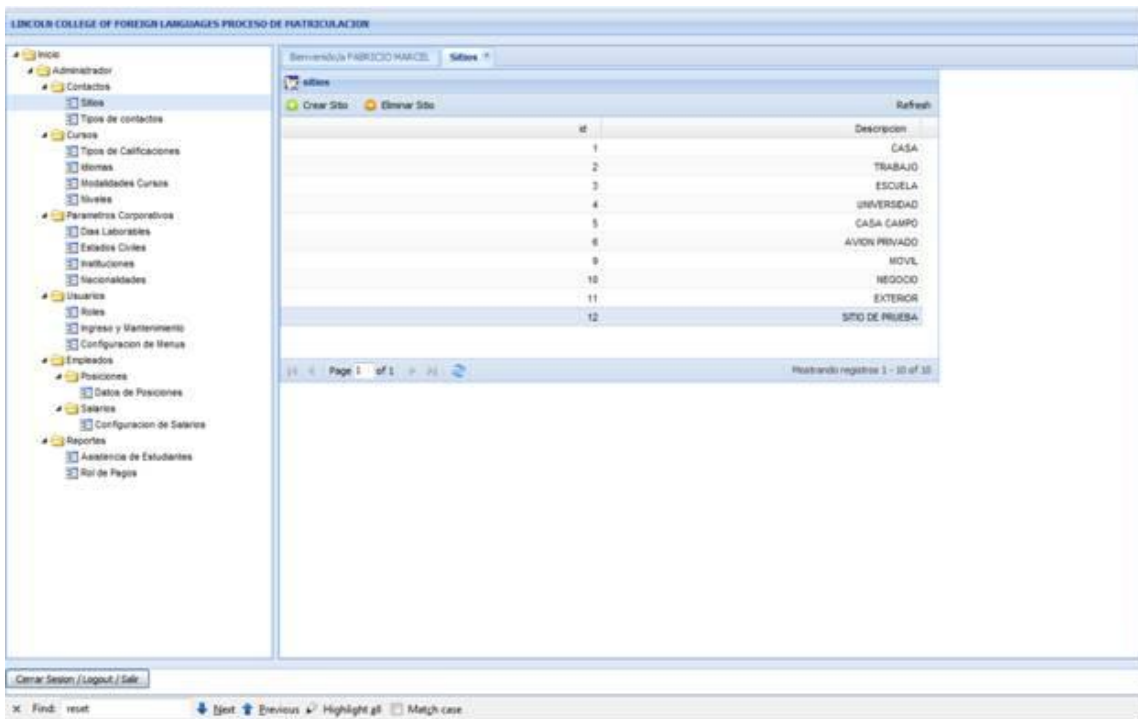
Page 1 of 1

Mostrando registros 1 - 9 of 9

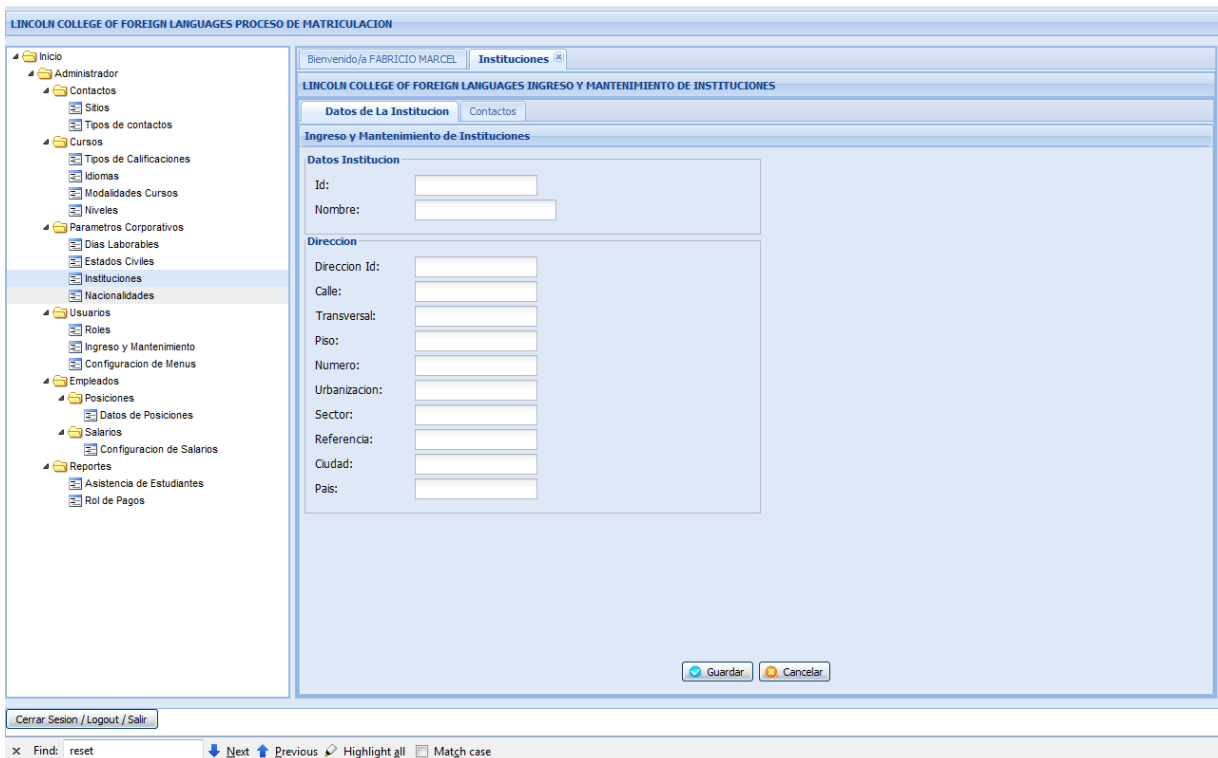
Cerrar Sesión / Logout / Salir

Find: reset Previous Highlight all Match case

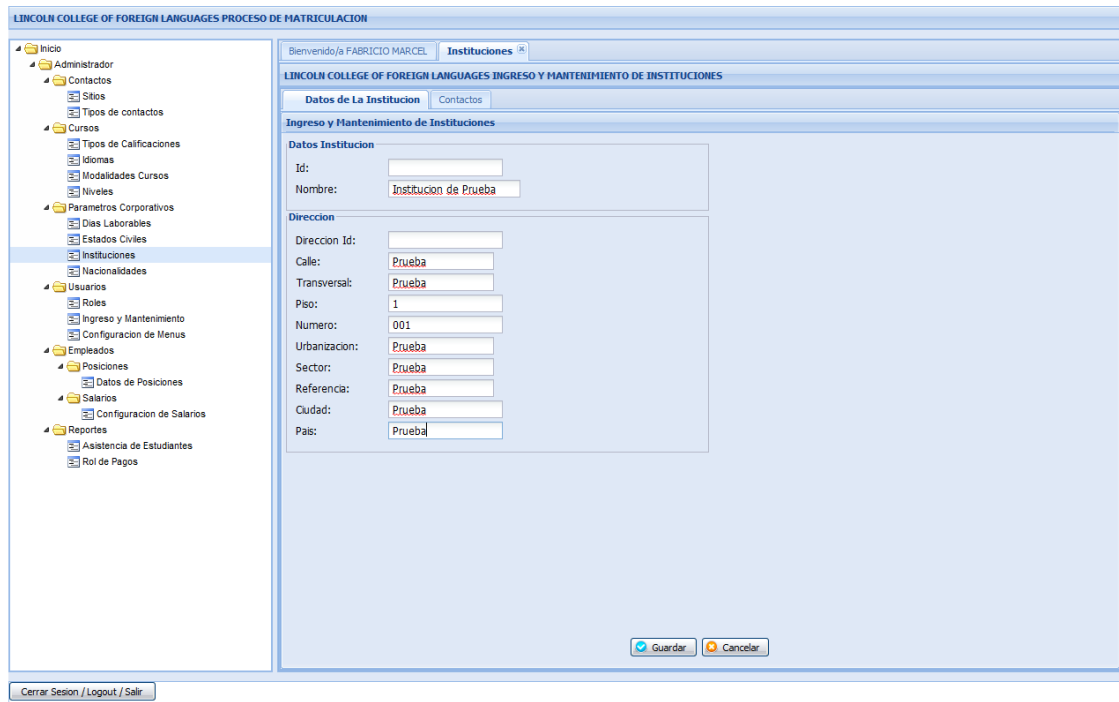
- Al quitar el enfoque de la celda donde se ingresó los datos , la tabla se refresca.



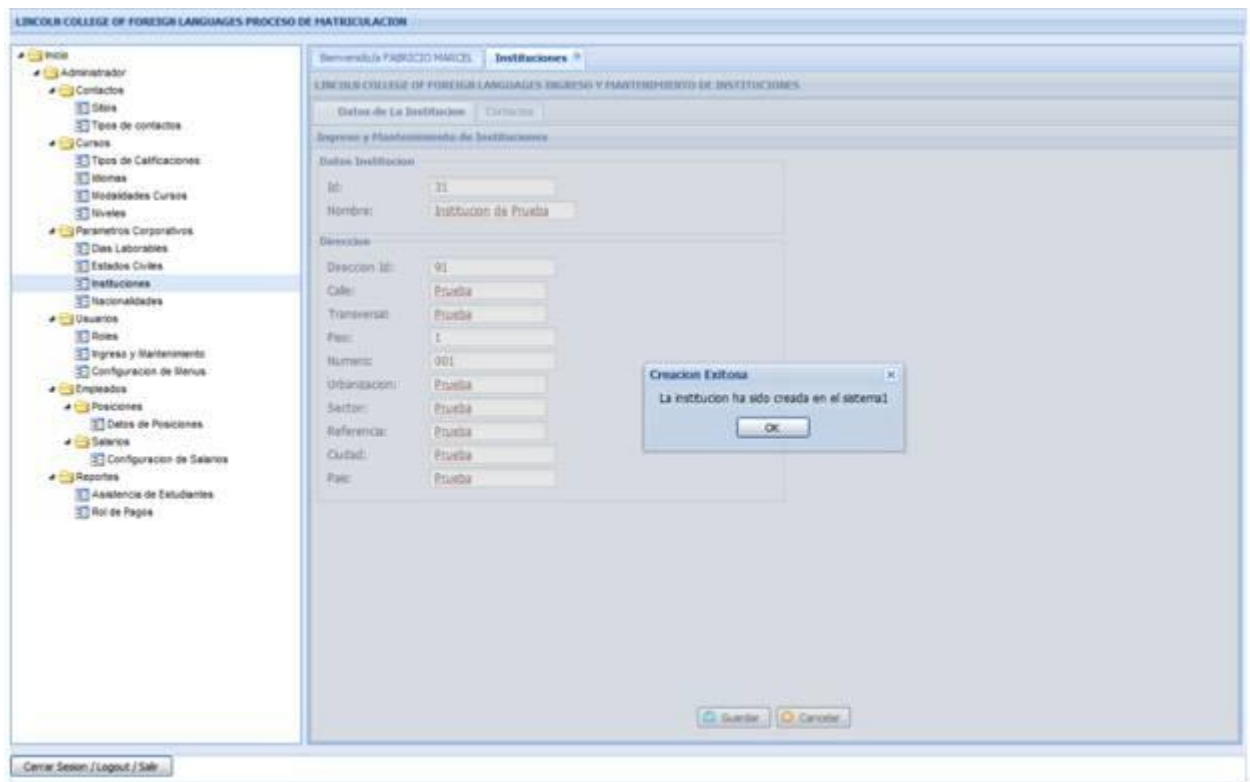
- Ahora seleccionamos la opción Instituciones del Viewport y aparece el formulario de ingreso de Instituciones.



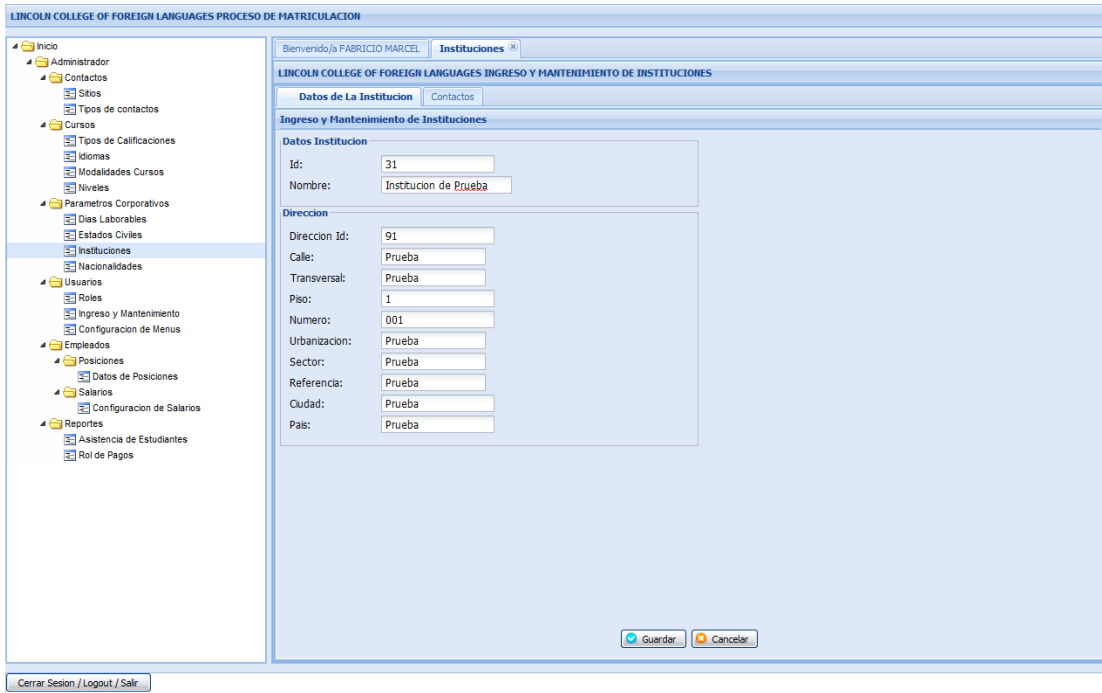
- Ingresamos “Institución de Prueba” en la casilla Nombre y “Prueba” en la casillas Calle , Transversal, Urbanización , Sector , Referencia , Ciudad , País , 1 en la casilla Piso y “001” en la casilla Numero.



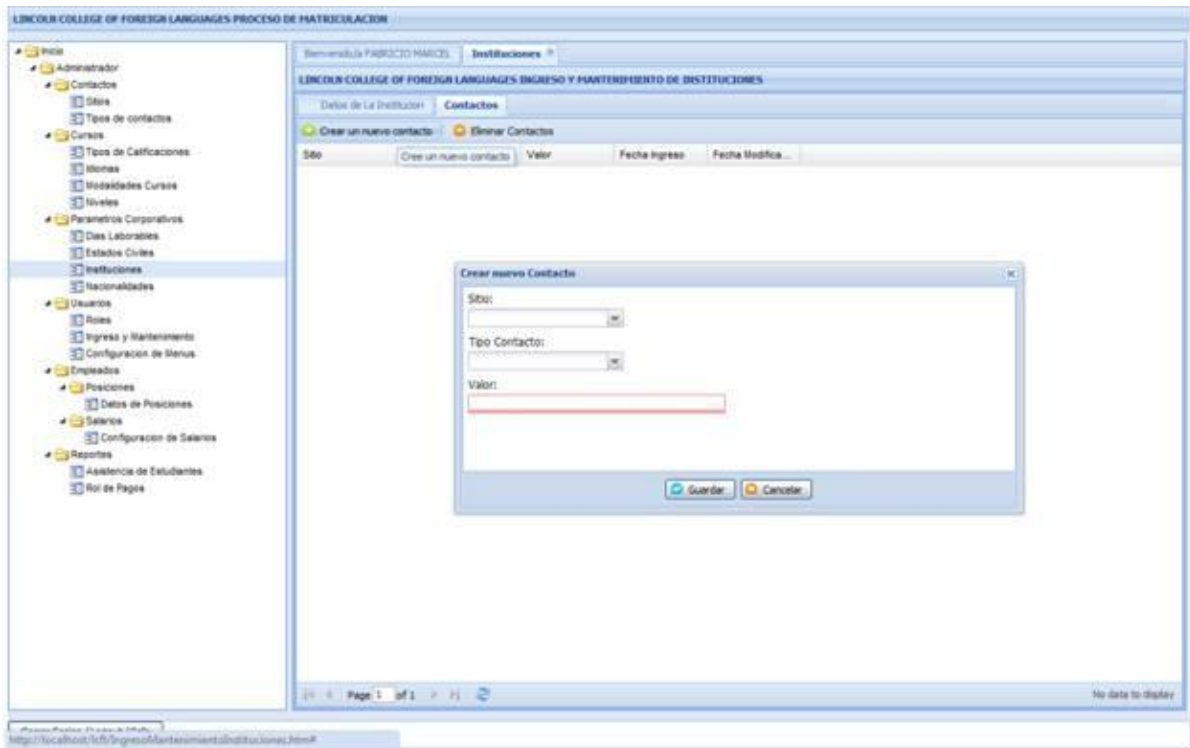
- Presionamos el botón Guardar y aparece la siguiente ventana de dialogo.



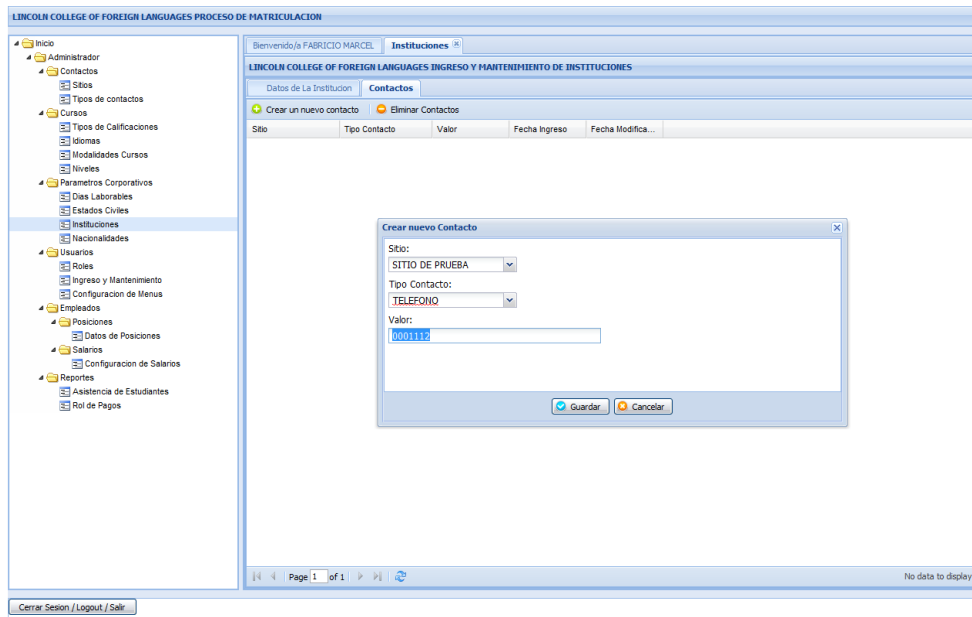
- Al presionar el botón OK de la ventana de confirmación se puede observar el campo Id tiene un numero asignado , en este caso el número 31.



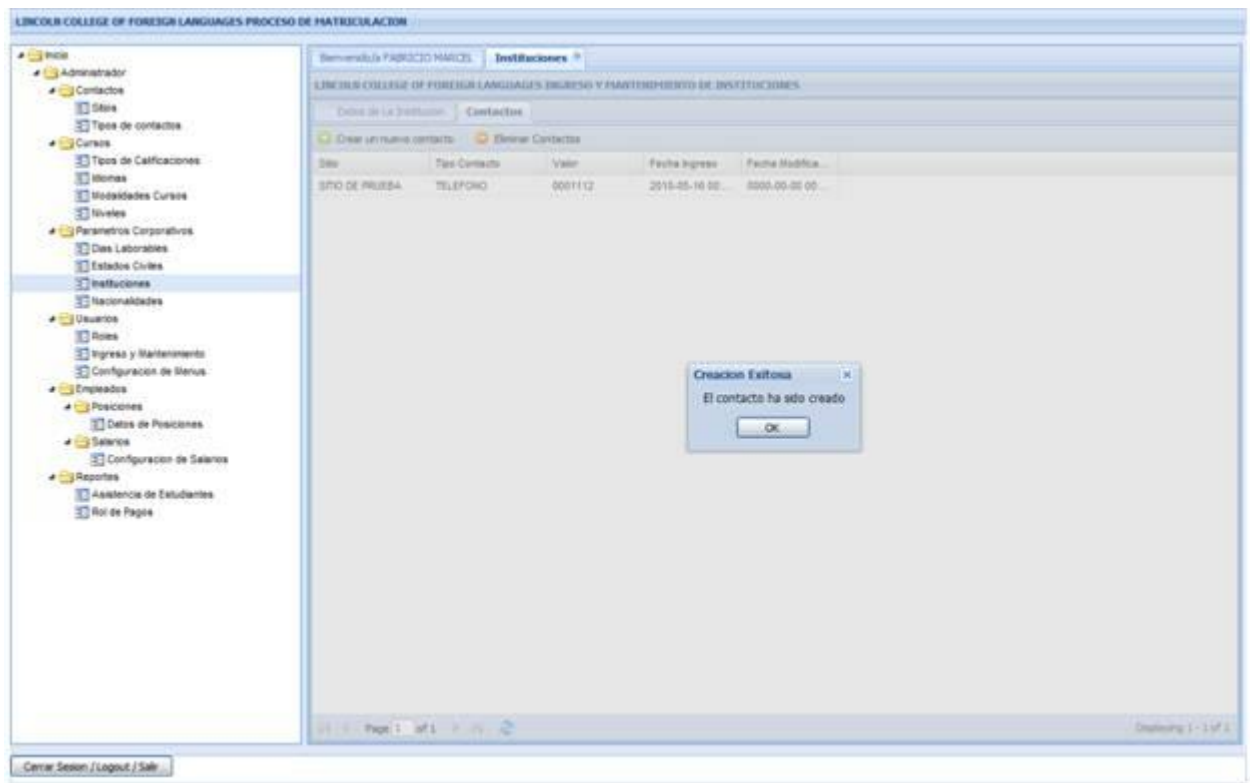
- Ahora procedemos a crear los contactos de la institución recién ingresada.
- Pulsando en la sección Contactos , presionamos el botón Crear un nuevo contacto

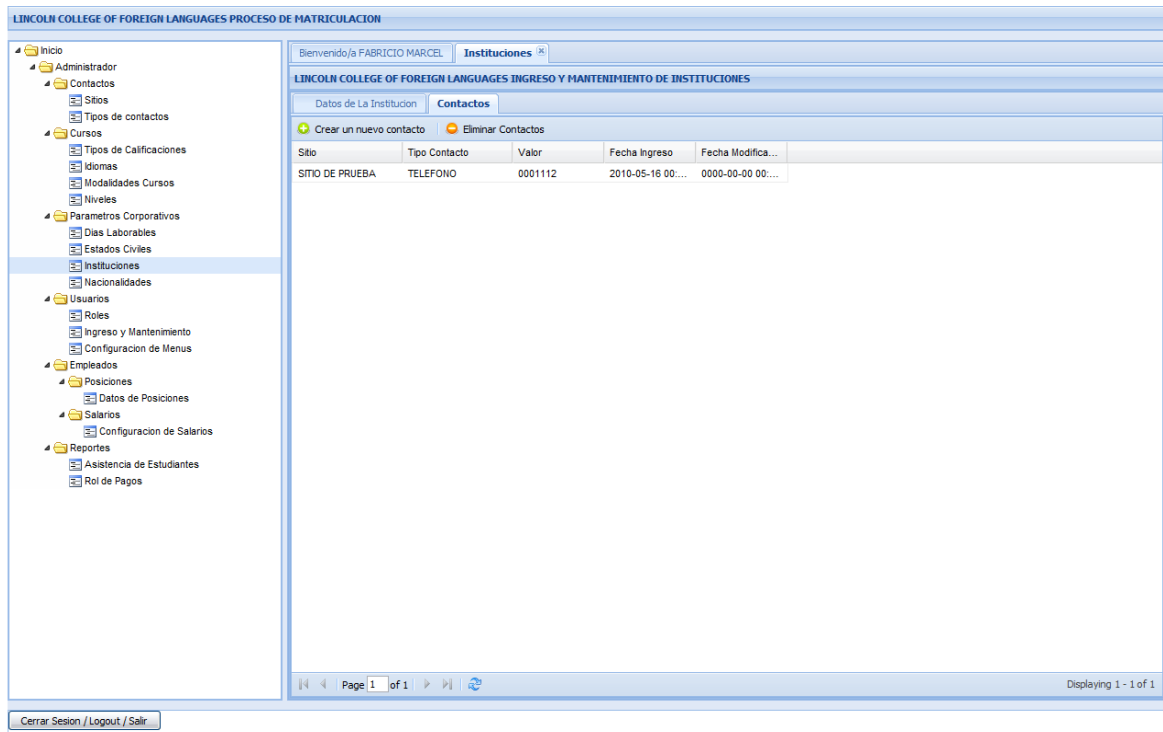


- Seleccionamos e Ingresamos los siguientes datos: Sitio:SITIO DE PRUEBA , Tipo Contacto: TELEFONO , Valor: 0001112



- Presionamos el botón Guardar para desplegar una ventana de dialogo y el nuevo contacto aparece en la tabla





Resultado esperado

El sistema ingresa nuevos datos de parámetros corporativos al sistema.

Evaluación de la Prueba

Prueba superada con éxito

Prueba 2: Comprobar que el sistema permite el mantenimiento de los parámetros corporativos.

Descripción

Ingresamos al sistema como Administrador , accediendo a la funcionalidad de Tipo de Contactos .

Condiciones de ejecución

Las condiciones de ejecución del caso de prueba son que el sistema permita actualizar un parámetro corporativo.

Entrada

- Seleccionamos la opción “Tipo de Contactos” del Viewport.

The screenshot displays the 'Tipos de contactos' (Contact Types) management interface. The left sidebar shows a navigation menu with 'Tipos de contactos' selected. The main content area features a table with the following data:

id	Descripcion
1	TELEFONO
2	CELULAR
3	FAX
4	BEEPER
5	PAGER
6	EMAIL
7	WEBSITE

At the bottom of the table, the record for 'WEBSITE' (id 7) is highlighted. The interface also includes a 'Crear Tipo de Contacto' button, an 'Eliminar Tipo de contacto' button, and a 'Refresh' button. The status bar at the bottom indicates 'Mostrando registros 1 - 7 of 7'.

- Hacemos doble clic sobre el parámetro “WEBSITE” e introducimos “SITIO WEB”

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL Tipos de contactos

Tipos de contactos

Crear Tipo de Contacto Eliminar Tipo de contacto Refresh

id	Descripcion
1	TELEFONO
2	CELULAR
3	FAX
4	BEEPER
5	PAGER
6	EMAIL
7	SITIO WEB

Page 1 of 1 Mostrando registros 1 - 7 of 7

Cerrar Sesion / Logout / Salir

- Al retirar el enfoque , la tabla se actualiza

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL Tipos de contactos

Tipos de contactos

Crear Tipo de Contacto Eliminar Tipo de contacto Refresh

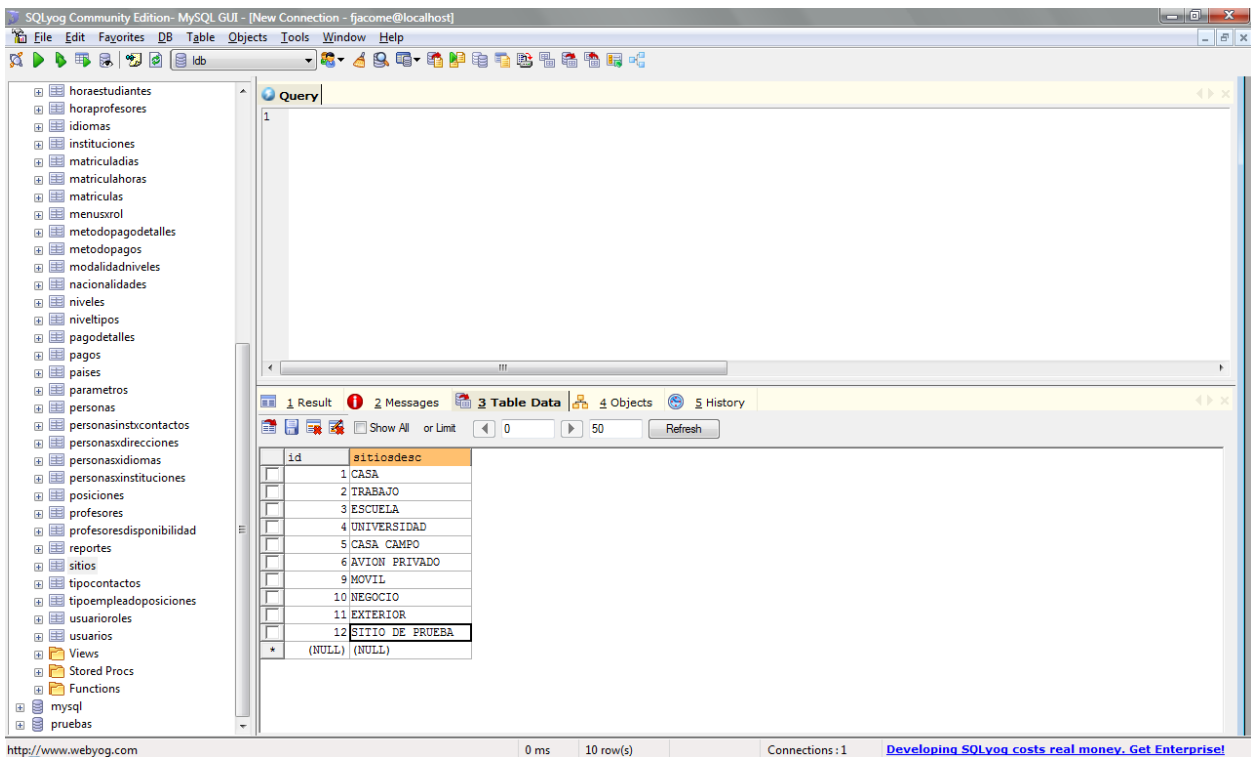
id	Descripcion
1	TELEFONO
2	CELULAR
3	FAX
4	BEEPER
5	PAGER
6	EMAIL
7	SITIO WEB

Page 1 of 1 Mostrando registros 1 - 7 of 7

http://localhost/lcfl/tipocontactos_EditorGrid.htm#

Resultado esperado

El sistema actualiza correctamente la modificación efectuada.



Evaluación de la Prueba

Prueba superada con éxito

Prueba 3: Comprobar que el sistema aplica integridad referencial.

Descripción

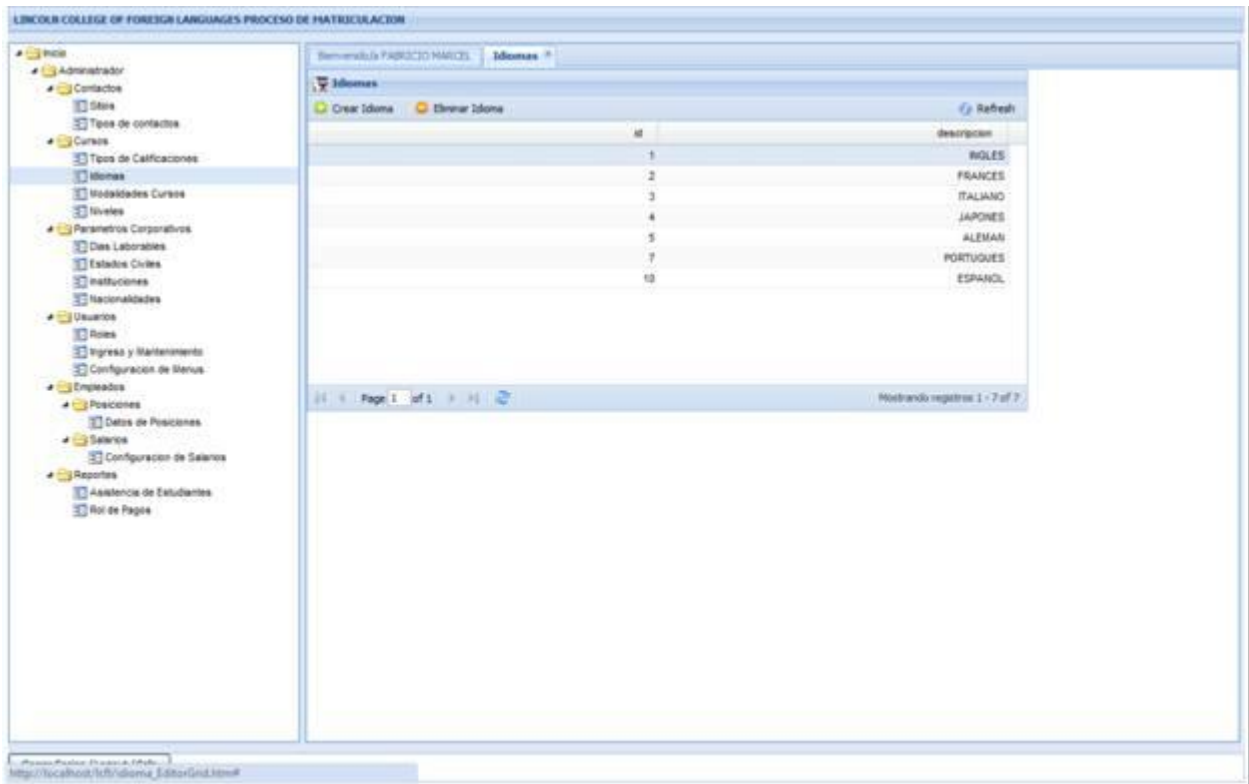
Ingresamos al sistema como Administrador, accediendo a la opción Idiomas.

Condiciones de ejecución

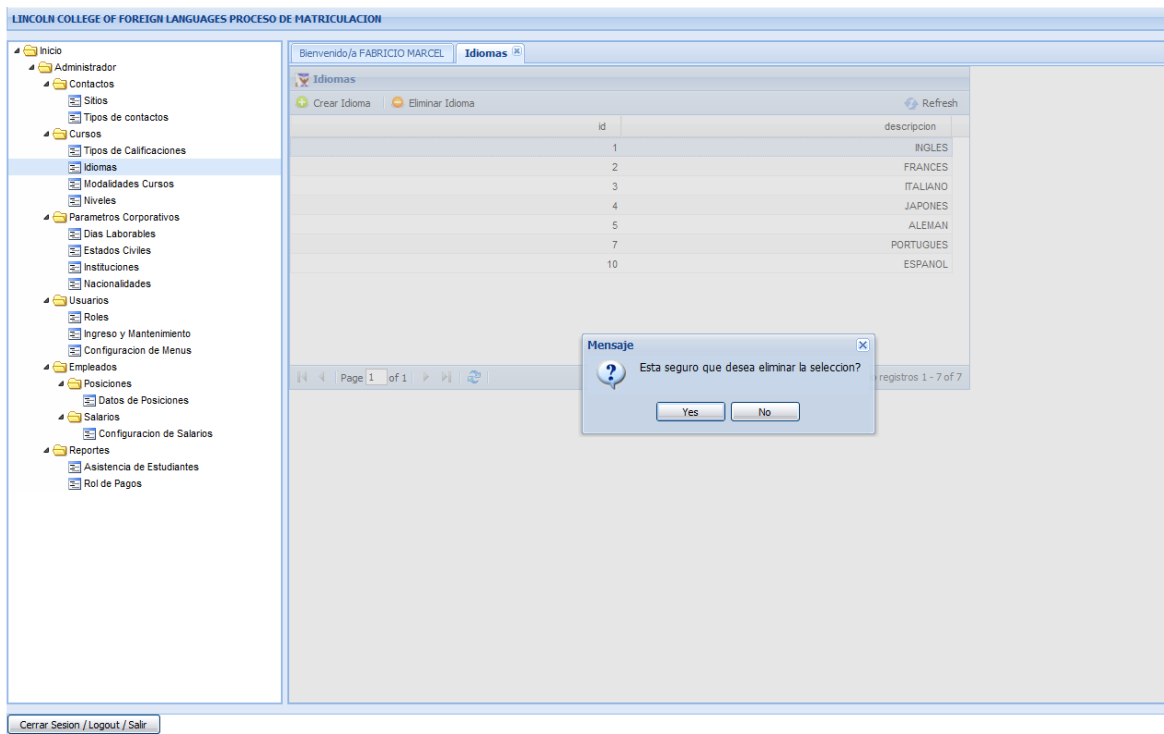
Las condiciones de ejecución del caso de prueba son que el sistema no realice una eliminación de un parámetro corporativo que está presente en otras tablas.

Entrada

- Seleccionamos la opción “Idiomas “del Viewport.



- Hacemos doble clic sobre el parámetro “INGLES” y presionamos el botón Eliminar Idioma, tras lo cual aparece una ventana de dialogo de confirmación.



- Presionamos Si y nos aparece el siguiente mensaje:”No se puede efectuar la eliminación, Existen Registros Hijos”

The screenshot shows a web application interface for 'LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION'. The user is logged in as 'FABRICIO MARCEL'. The main content area displays a table of languages with columns 'id' and 'descripcion'. The table contains 7 rows of data. A modal dialog box is overlaid on the table, displaying the error message: 'No se puede efectuar la eliminacion!!! Existen registros hijos'. The dialog has an 'OK' button. The table data is as follows:

id	descripcion
1	INGLES
2	FRANCES
3	ITALIANO
4	JAPONES
5	ALEMAN
7	PORTUGUES
10	ESPANOL

Resultado esperado

El sistema aplica la integridad referencial para evitar tener registros huérfanos.

Evaluación de la Prueba

Prueba superada con éxito

FASE: CONSTRUCCIÓN	ID: CNS004
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de Prueba: Gestionar Calificaciones
VERSIÓN:	1.0

Descripción: Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Gestionar Calificaciones” .

Se ingresa o actualiza datos con el perfil de Profesor.

Comprobar que se puede ingresar una o más columnas de aportes a un cuadro de calificaciones

Descripción

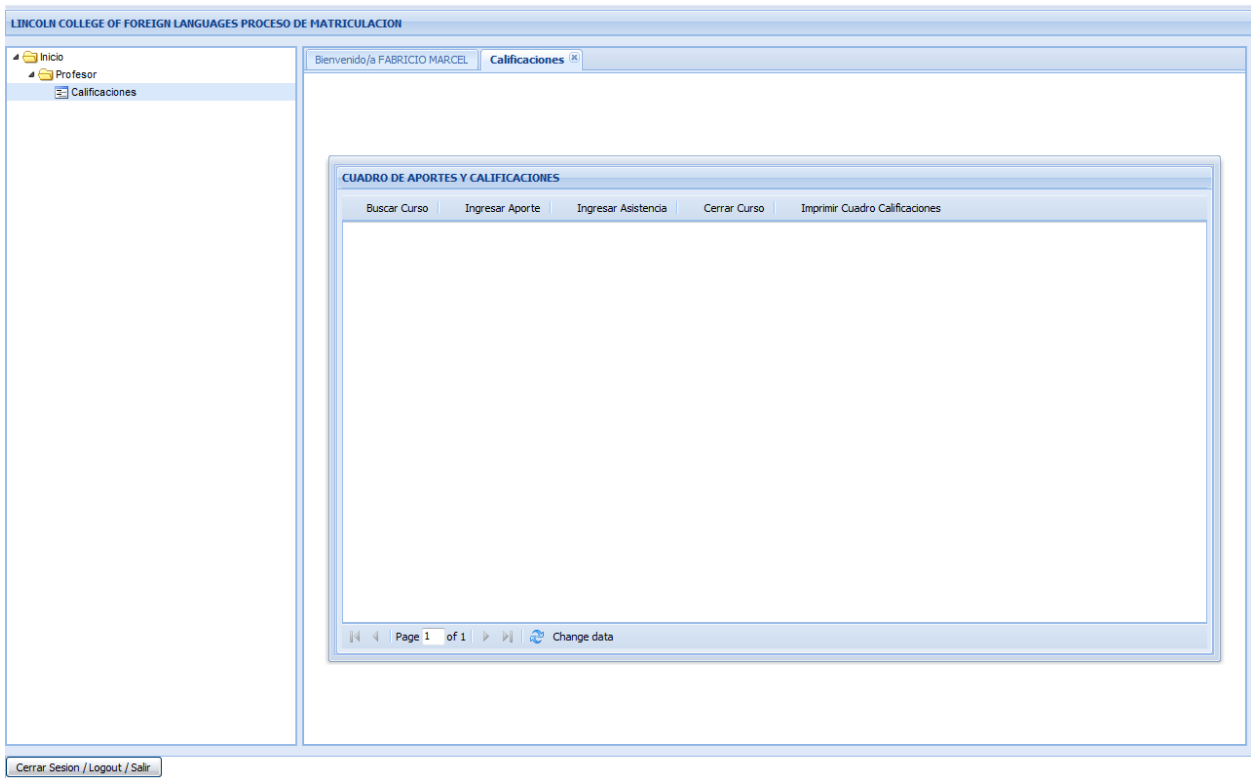
Ingresamos al sistema como Profesor para ingresar columnas de aportes a un cuadro de calificaciones.

Condiciones de ejecución

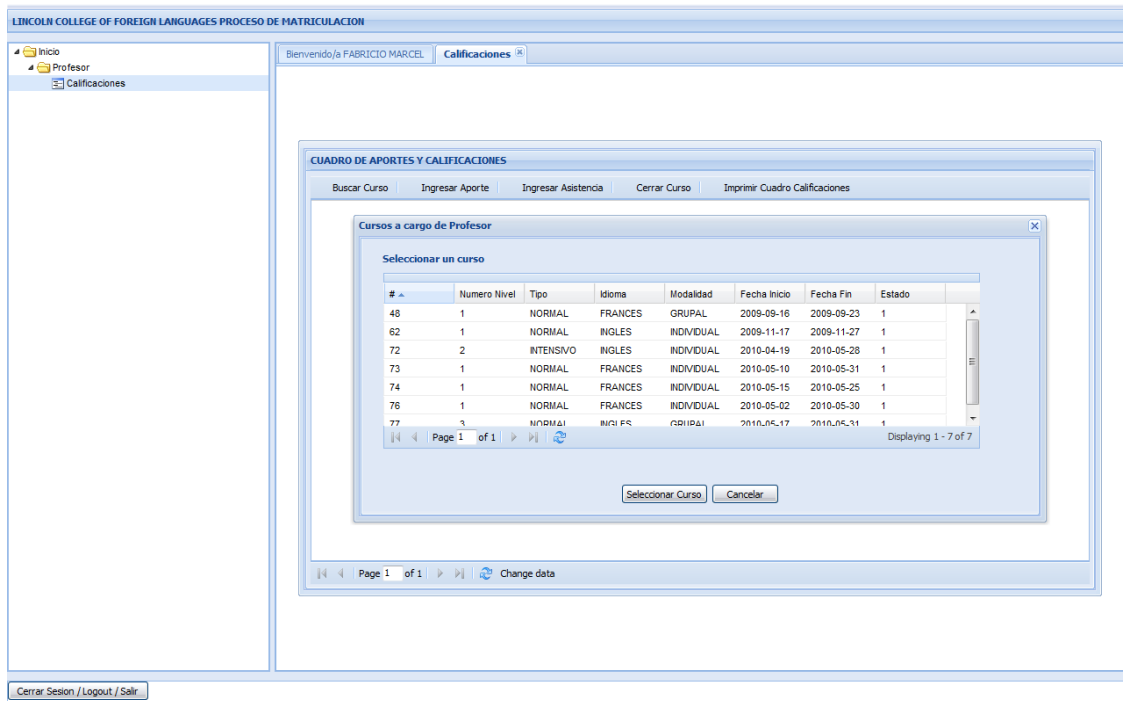
Las condiciones de ejecución del caso de prueba son que el sistema permita ingresar columnas de aportes en un cuadro de calificaciones

Entrada

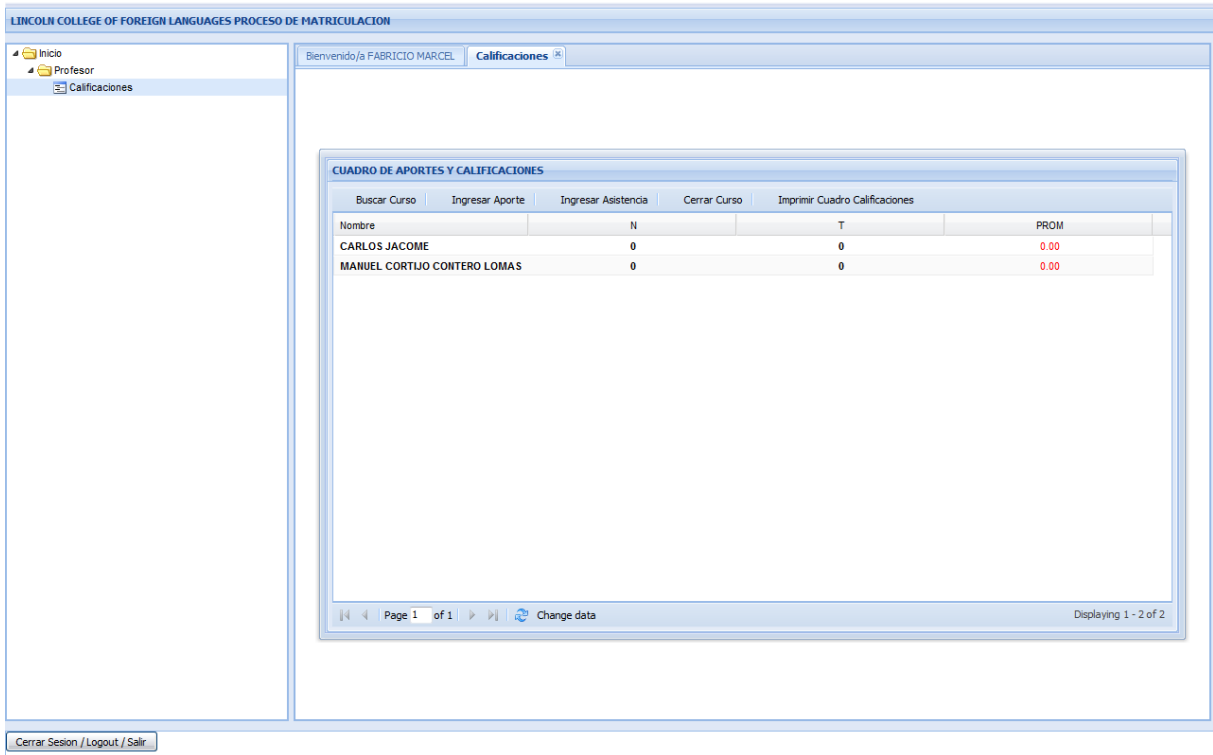
- Seleccionamos la opción “Calificaciones” del Viewport , lo cual despliega la siguiente pantalla



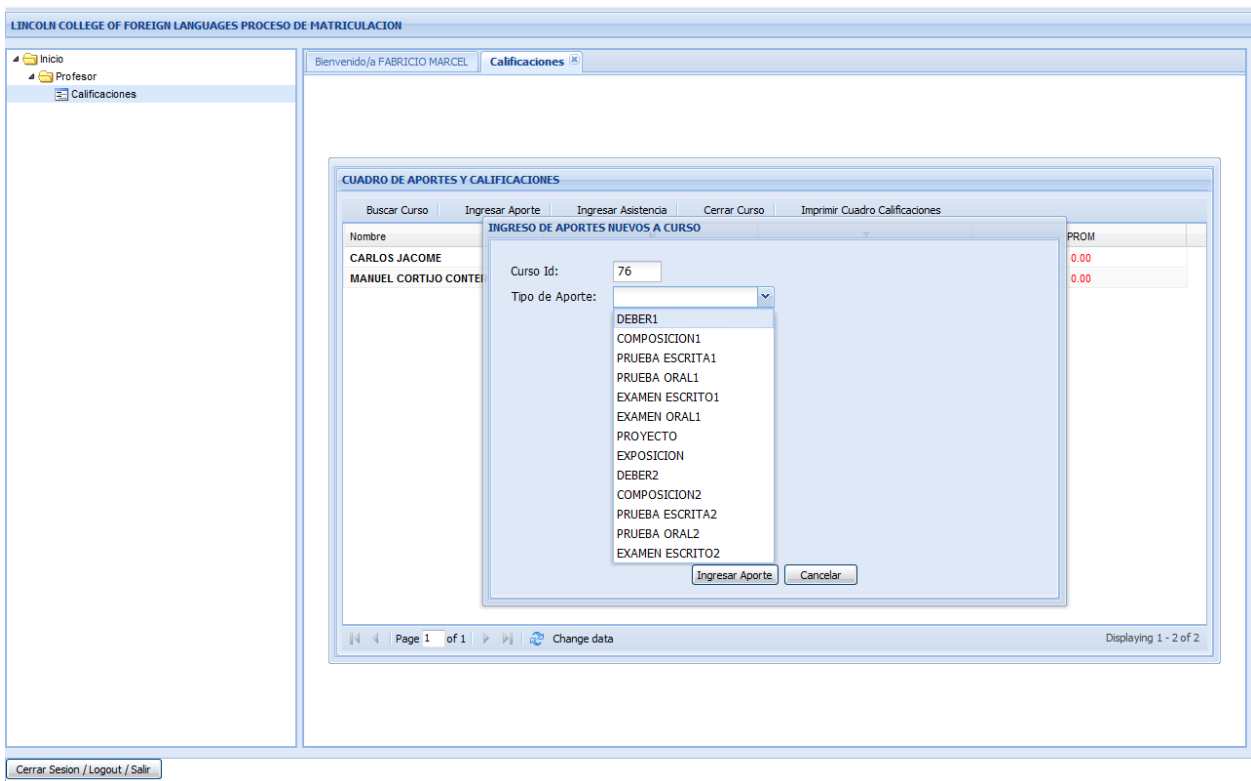
- Presionamos el botón “Buscar Curso”, y aparece una ventana con una lista de cursos asignados para desplegar el cuadro de calificaciones.



- Seleccionamos el curso con Id : 76 que creamos en el caso de prueba Matricular Alumnos y presionamos el botón Seleccionar Curso. Aparece el cuadro de calificaciones.



- Presionamos el botón Ingresar Aporte para ingresar una columna de aportes.



- Seleccionamos el aporte “DEBER1” y presionamos el botón Ingresar Aporte. Se crea D1

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL **Calificaciones**

Inicio
 Profesor
 Calificaciones

CUADRO DE APORTES Y CALIFICACIONES

Buscar Curso Ingresar Aporte Ingresar Asistencia Cerrar Curso Imprimir Cuadro Calificaciones

Nombre	D1	N	T	PROM
MANUEL CORTIJO CONTERO LOMAS	0	1	0	0.00
CARLOS JACOME	0	1	0	0.00

Page 1 of 1 Change data Displaying 1 - 2 of 2

Cerrar Sesión / Logout / Salir

- Ahora ingresamos otro aporte

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL **Calificaciones**

Inicio
 Profesor
 Calificaciones

CUADRO DE APORTES Y CALIFICACIONES

Buscar Curso Ingresar Aporte Ingresar Asistencia Cerrar Curso Imprimir Cuadro Calificaciones

INGRESO DE APORTES NUEVOS A CURSO

Curso Id: 76

Tipo de Aporte:

- DEBER1
- COMPOSICION1
- PRUEBA ESCRITA1
- PRUEBA ORAL1
- EXAMEN ESCRITO1
- EXAMEN ORAL1
- PROYECTO
- EXPOSICION
- DEBER2
- COMPOSICION2
- PRUEBA ESCRITA2
- PRUEBA ORAL2
- EXAMEN ESCRITO2

Ingresar Aporte Cancelar

N	T	PROM
1	0	0.00
1	0	0.00

Page 1 of 1 Change data Displaying 1 - 2 of 2

Cerrar Sesión / Logout / Salir

- Se crea el aporte C1

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL **Calificaciones**

Inicio
 Profesor
 Calificaciones

CUADRO DE APORTES Y CALIFICACIONES

Buscar Curso Ingresar Aporte Ingresar Asistencia Cerrar Curso Imprimir Cuadro Calificaciones

Nombre	D1	C1	N	T	PROM
MANUEL CORTIJO CONTERO LOMAS	0	0	2	0	0.00
CARLOS JACOME	0	0	2	0	0.00

Page 1 of 1 Change data Displaying 1 - 2 of 2

Cerrar Sesión / Logout / Salir

- Ahora ingresamos el valor “14.50” en la columna D1 o Deber1 y 15 en la columna C1 o Composicion1 de la persona “MANUEL CORTIJO CONTERO LOMAS”.

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL **Calificaciones**

Inicio
 Profesor
 Calificaciones

CUADRO DE APORTES Y CALIFICACIONES

Buscar Curso Ingresar Aporte Ingresar Asistencia Cerrar Curso Imprimir Cuadro Calificaciones

Nombre	D1	C1	N	T	PROM
MANUEL CORTIJO CONTERO LOMAS	14.5	15	2	29.5	14.75
CARLOS JACOME	0	0	2	0	0.00

Page 1 of 1 Change data Displaying 1 - 2 of 2

Cerrar Sesión / Logout / Salir

- Los aportes se mantienen en color rojo ya que no son valores suficientes para aprobación, aparece también el número de aportes bajo la columna N , la suma de los aportes bajo T y el promedio bajo PROM. El promedio de este caso es : $14.5+15=29.5$ con un promedio de 14.75 en rojo ya que no es un valor de aprobación.
- Ahora ingresamos el 18 para la columna D1 y 16.80 para la columna C1 del alumno “CARLOS JACOME”

Nombre	D1	C1	N	T	PROM
MANUEL CORTUJO CONTERO LOMAS	14.5	15	2	29.5	14.75
CARLOS JACOME	18	16.8	2	34.8	17.40

- Los aportes están en color verde ya que son valores mayores a 16 , que es el valor mínimo para aprobación. Además , la suma es de $18 + 16.80 = 34.80$ para un promedio de 17.40 en verde ya que se trata de un valor de aprobación , es decir mayor a 16/20.

Resultado esperado

El sistema crea las columnas de aportes exitosamente , permite el ingreso y modifica correctamente los aportes de calificaciones.

Evaluación de la Prueba

Prueba superada con éxito

FASE: CONSTRUCCIÓN	ID: CNS005
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de Prueba: Ingresar Asistencia
VERSIÓN:	1.0

Descripción

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ingresar Asistencia”.

Se ingresa o actualiza datos con el perfil de Profesor.

Comprobar que se puede ingresar la asistencia en línea de los alumnos de un curso

Descripción

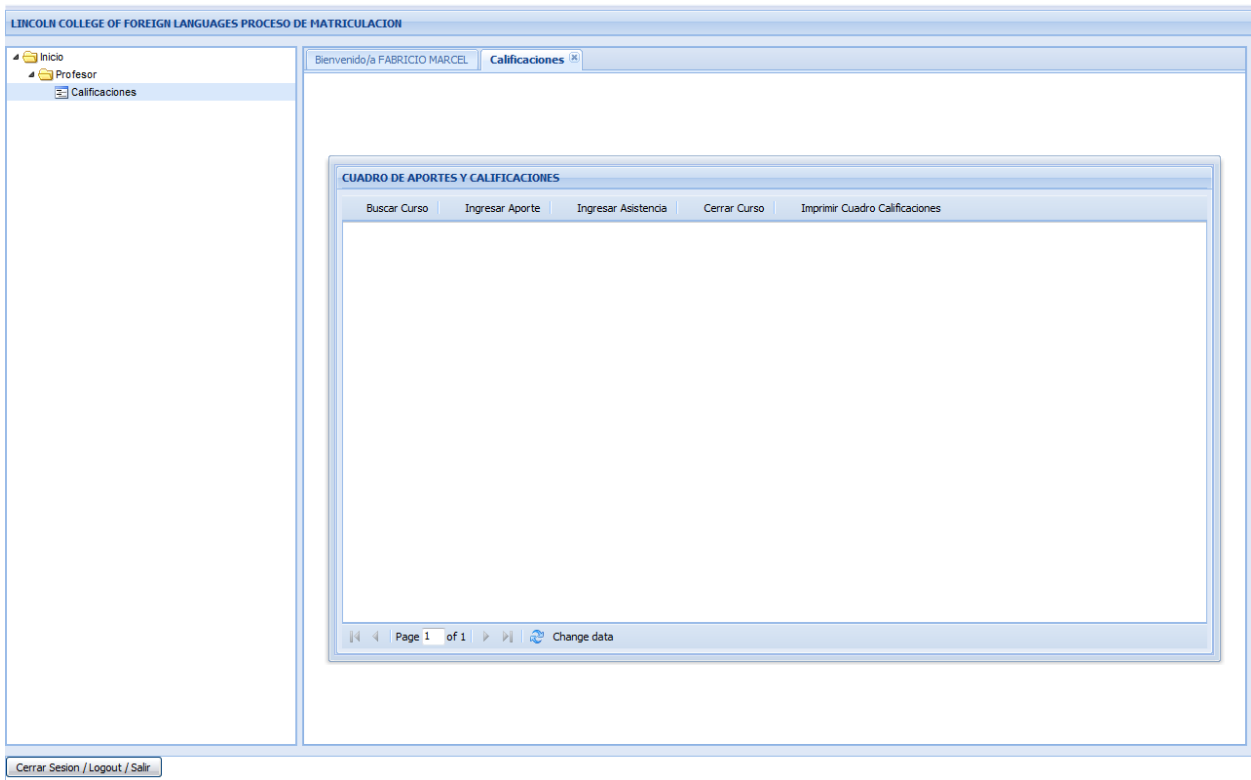
Ingresamos al sistema como Profesor para ingresar la asistencia de un curso.

Condiciones de ejecución

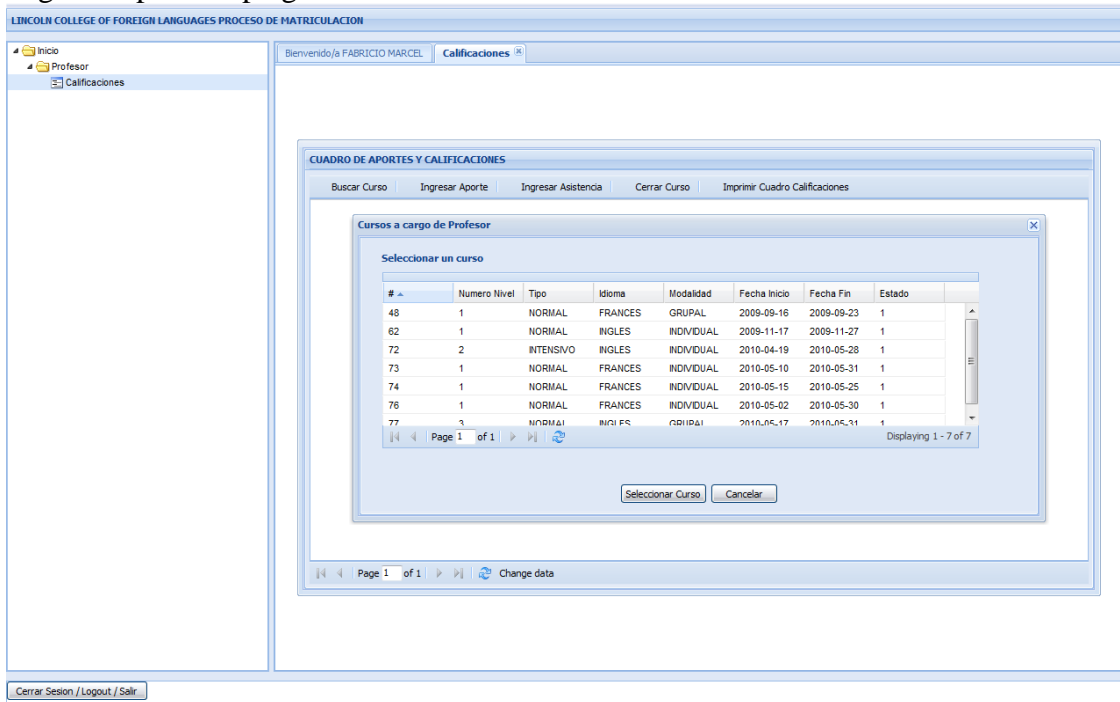
Las condiciones de ejecución del caso de prueba son que el sistema ingrese la asistencia del día para un curso.

Entrada

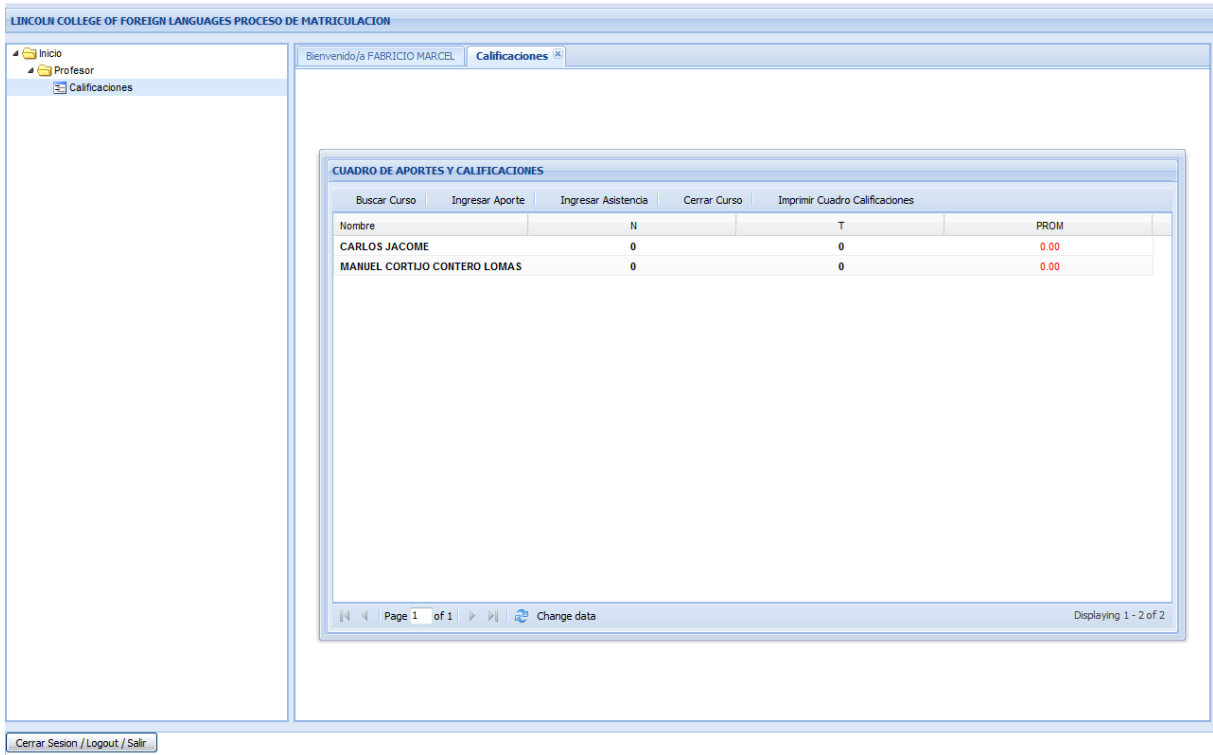
- Seleccionamos la opción “Calificaciones” del Viewport , lo cual despliega la siguiente pantalla



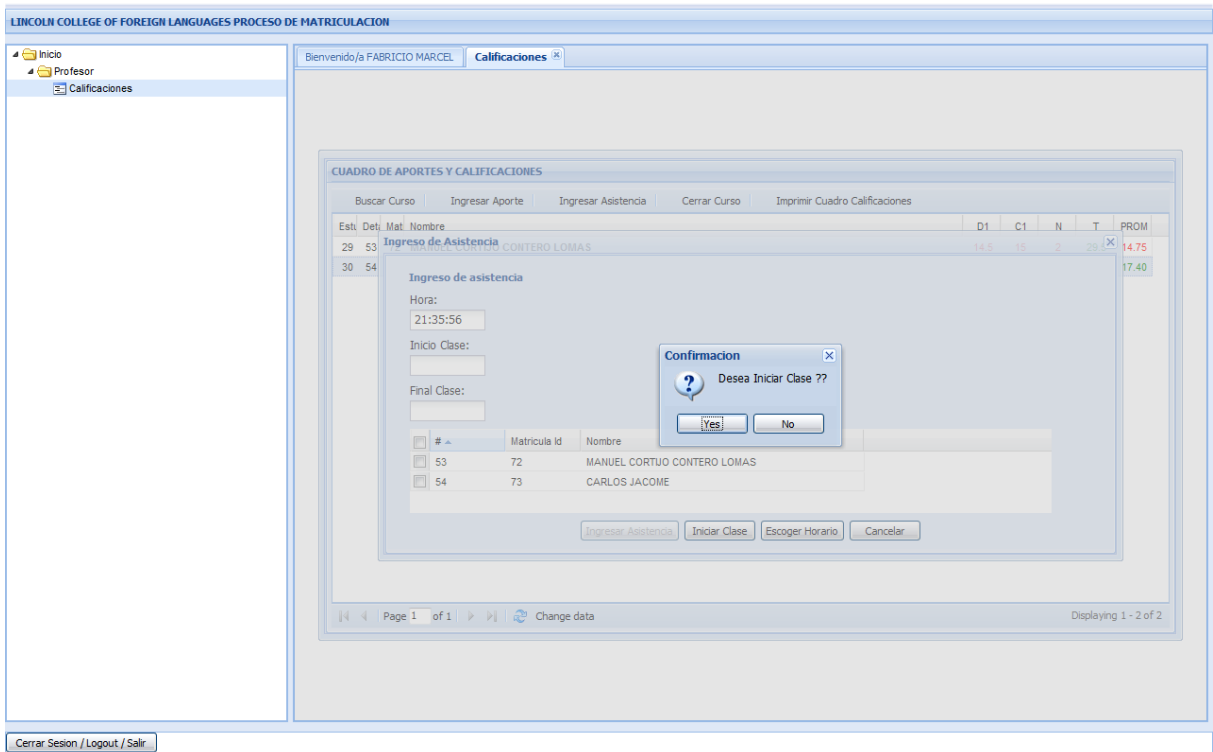
- Presionamos el botón “Buscar Curso”, y aparece una ventana con una lista de cursos asignados para desplegar el cuadro de calificaciones.



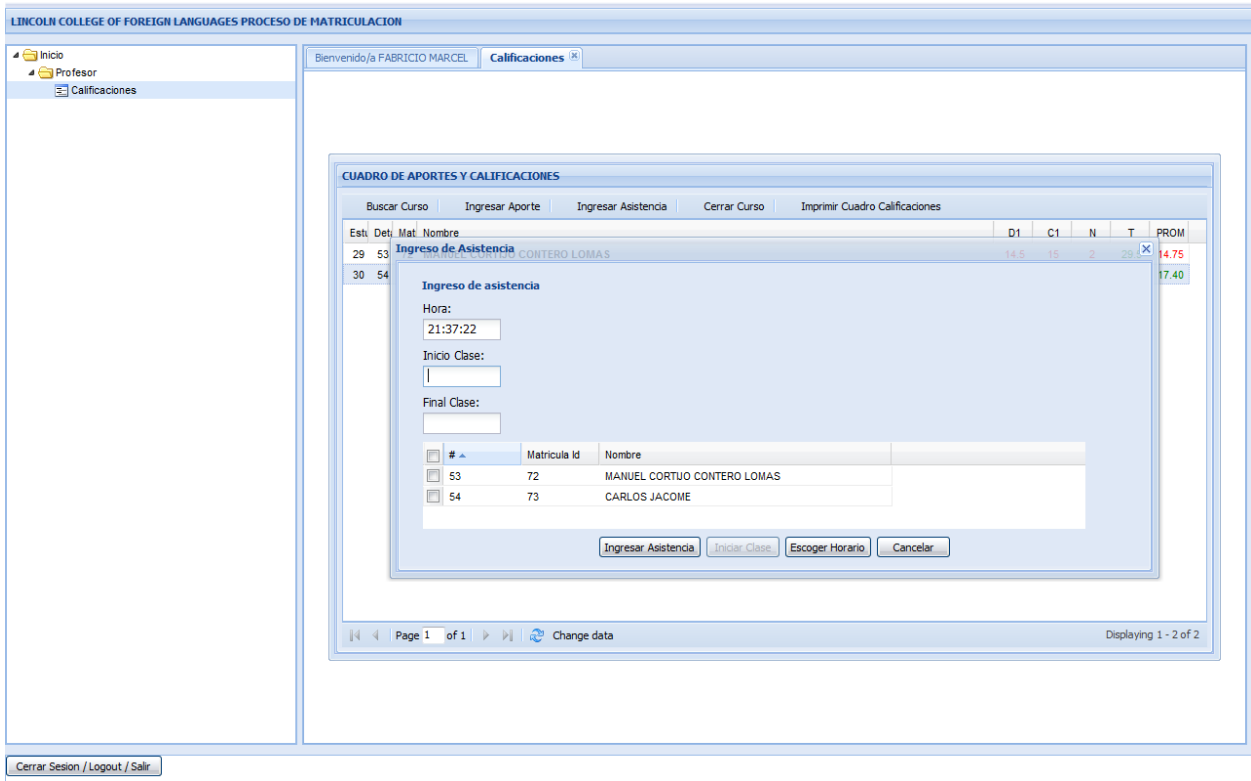
- Seleccionamos el curso con Id : 76 que creamos en el caso de prueba Matricular Alumnos y presionamos el botón Seleccionar Curso. Aparece el cuadro de calificaciones.



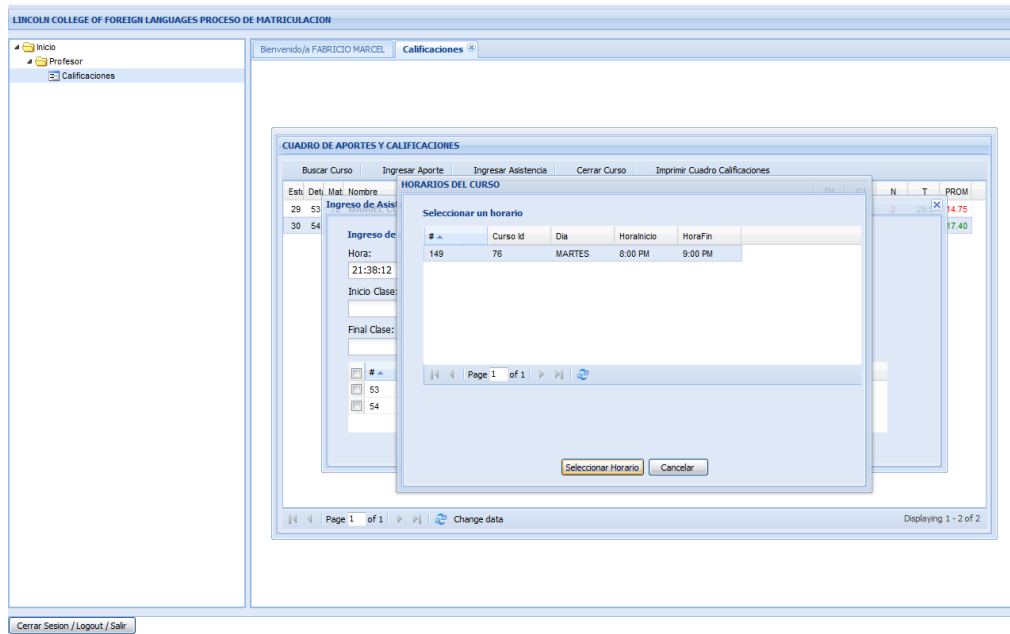
- Presionamos el botón Ingresar Asistencia y aparece la ventana de ingreso de asistencia.
- Presionamos el botón Iniciar Clase y se nos despliega una ventana de confirmación.



- Al aceptar la confirmación se activa el botón Ingresar Asistencia que anteriormente estaba deshabilitado.



- Presionamos el botón Escoger Horario para escoger el horario para el cual se va a ingresar la asistencia, recordemos que un curso puede tener varios horarios. Se despliega una ventana en donde escogemos un horario y presionamos el botón Seleccionar Horario.



LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL **Calificaciones**

Inicio
 Profesor
 Calificaciones

CUADRO DE APORTES Y CALIFICACIONES

Buscar Curso Ingresar Aporte Ingresar Asistencia Cerrar Curso Imprimir Cuadro Calificaciones

Est.	Det.	Mat.	Nombre	D1	C1	N	T	PROM
29	53		Ingreso de Asistencia CONTERO LOMAS	14.5	15	2	29	14.75
30	54		Ingreso de asistencia					17.40

Hora: 21:40:19

Inicio Clase: 8:00 PM

Final Clase: 9:00 PM

#	Matricula Id	Nombre
<input type="checkbox"/>	53	72 MANUEL CORTUO CONTERO LOMAS
<input type="checkbox"/>	54	73 CARLOS JACOME

Ingresar Asistencia Iniciar Clase Escoger Horario Cancelar

Page 1 of 1 Change data Displaying 1 - 2 of 2

Cerrar Sesión / Logout / Salir

- Seleccionamos los alumnos que se encuentran presentes

LINCOLN COLLEGE OF FOREIGN LANGUAGES PROCESO DE MATRICULACION

Bienvenido/a FABRICIO MARCEL **Calificaciones**

Inicio
 Profesor
 Calificaciones

CUADRO DE APORTES Y CALIFICACIONES

Buscar Curso Ingresar Aporte Ingresar Asistencia Cerrar Curso Imprimir Cuadro Calificaciones

Est.	Det.	Mat.	Nombre	D1	C1	N	T	PROM
29	53		Ingreso de Asistencia CONTERO LOMAS	14.5	15	2	29	14.75
30	54		Ingreso de asistencia					17.40

Hora: 21:40:38

Inicio Clase: 8:00 PM

Final Clase: 9:00 PM

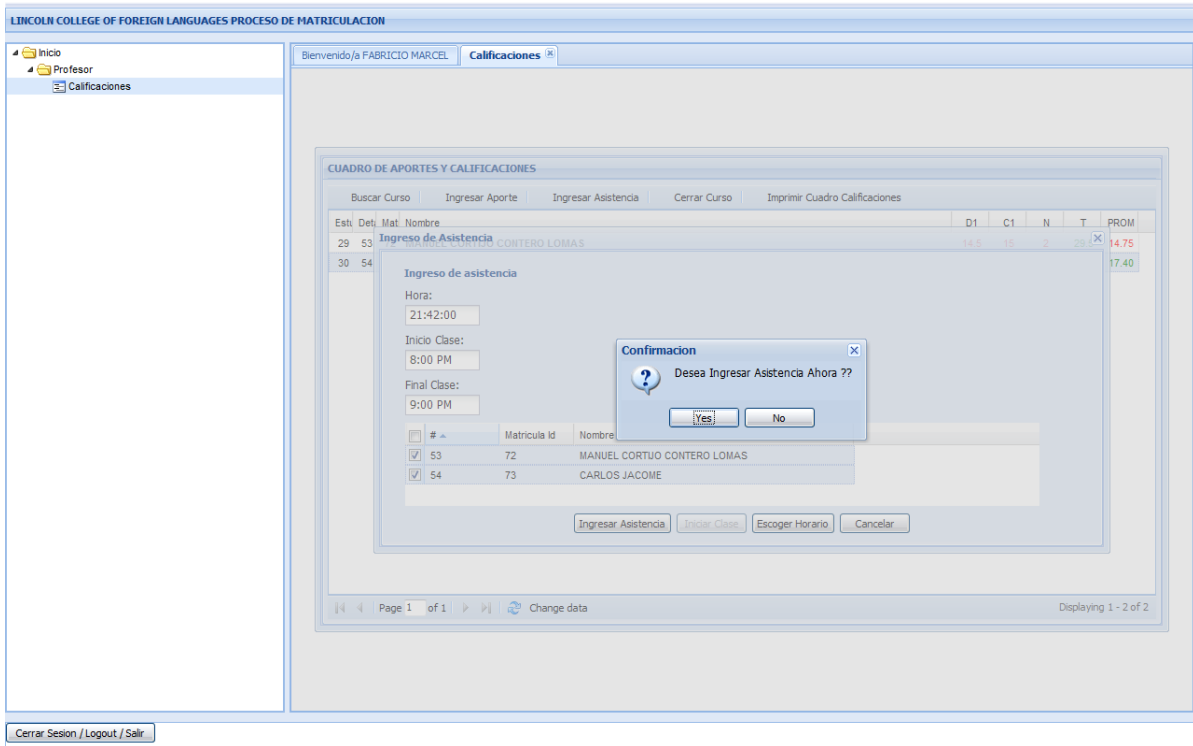
#	Matricula Id	Nombre
<input checked="" type="checkbox"/>	53	72 MANUEL CORTUO CONTERO LOMAS
<input checked="" type="checkbox"/>	54	73 CARLOS JACOME

Ingresar Asistencia Iniciar Clase Escoger Horario Cancelar

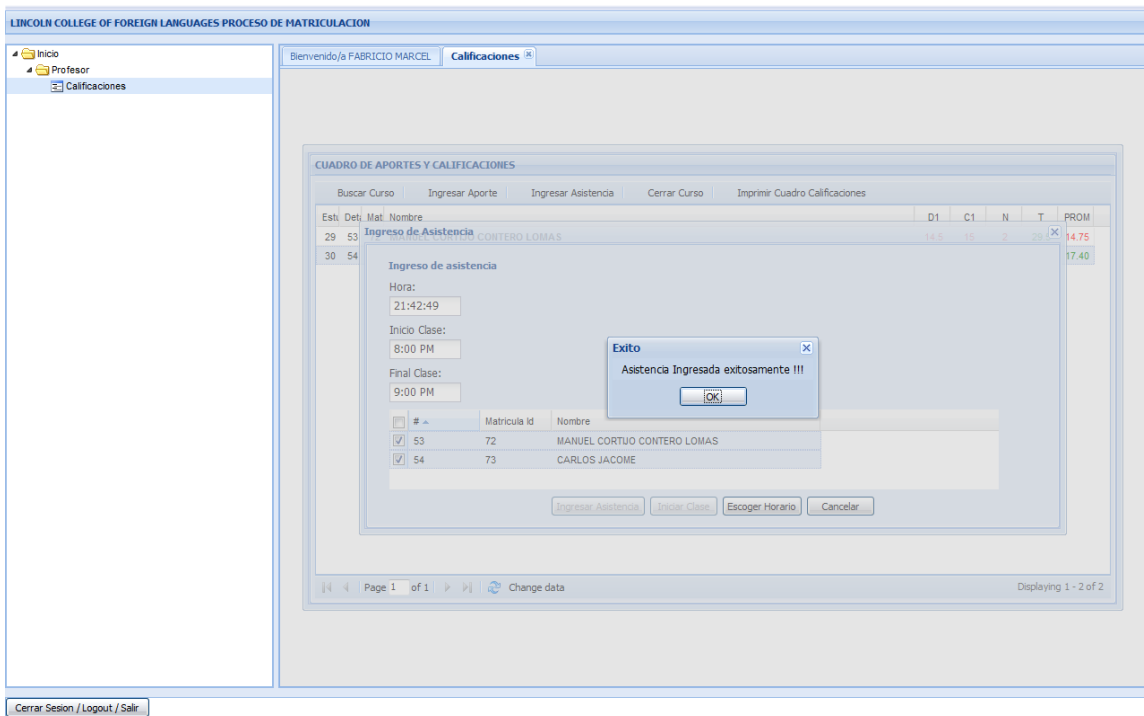
Page 1 of 1 Change data Displaying 1 - 2 of 2

Cerrar Sesión / Logout / Salir

- Ahora presionamos el botón Ingresar Asistencia . Aparece una ventana de confirmación.



- Al aceptar la confirmación aparece una ventana de éxito.



- Presionamos OK y cerramos la ventana de Ingreso de Asistencia donde podemos ver que el botón Ingresar Asistencia esta deshabilitado para evitar modificaciones o ingresar la asistencia dos veces.

Resultado esperado

El sistema permite el ingreso de la asistencia del curso en línea.

Evaluación de la Prueba

Prueba superada con éxito.

FASE: CONSTRUCCIÓN	ID: CNS006
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de Prueba: Generar Reporte de Asistencia
VERSIÓN:	1.0

Descripción

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Generar Reporte de Asistencia”.

Comprobar que se pueda generar un reporte de asistencia de alumnos a los cursos dentro de un mes en particular.

Descripción

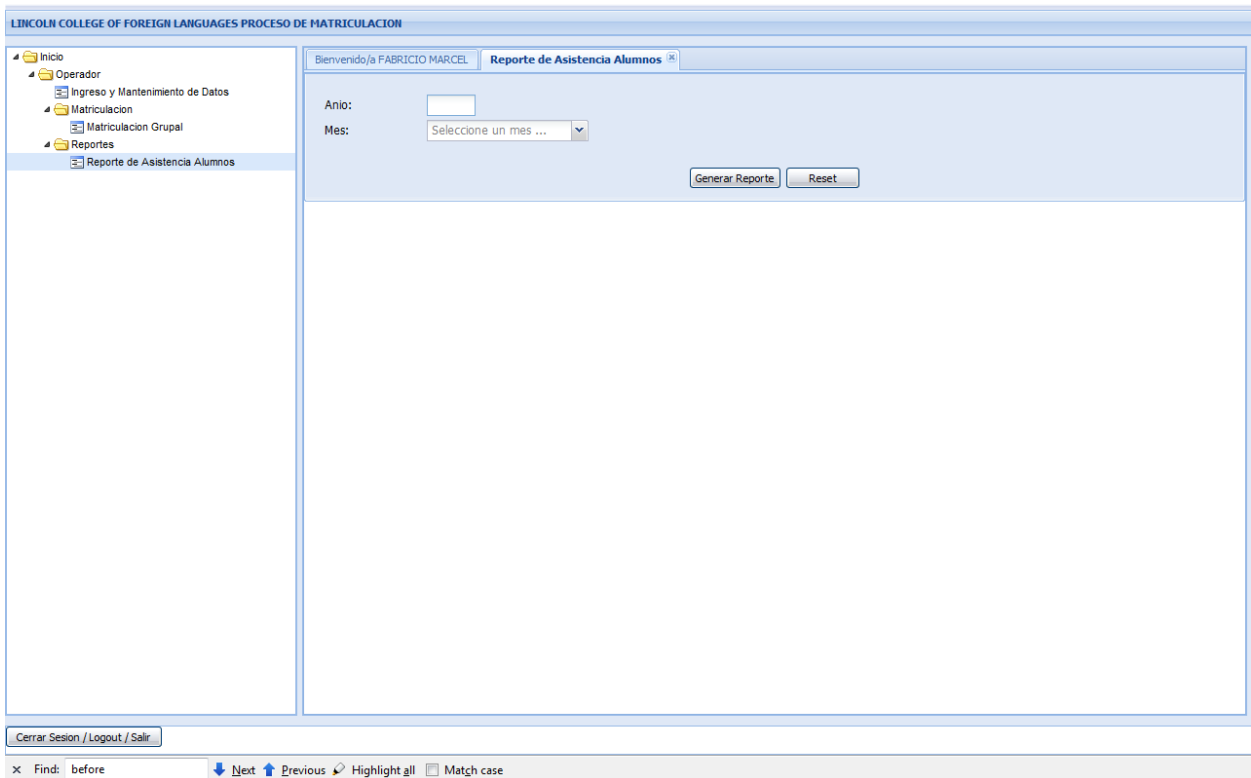
Ingresamos al sistema como Operador para generar un reporte de asistencia.

Condiciones de ejecución

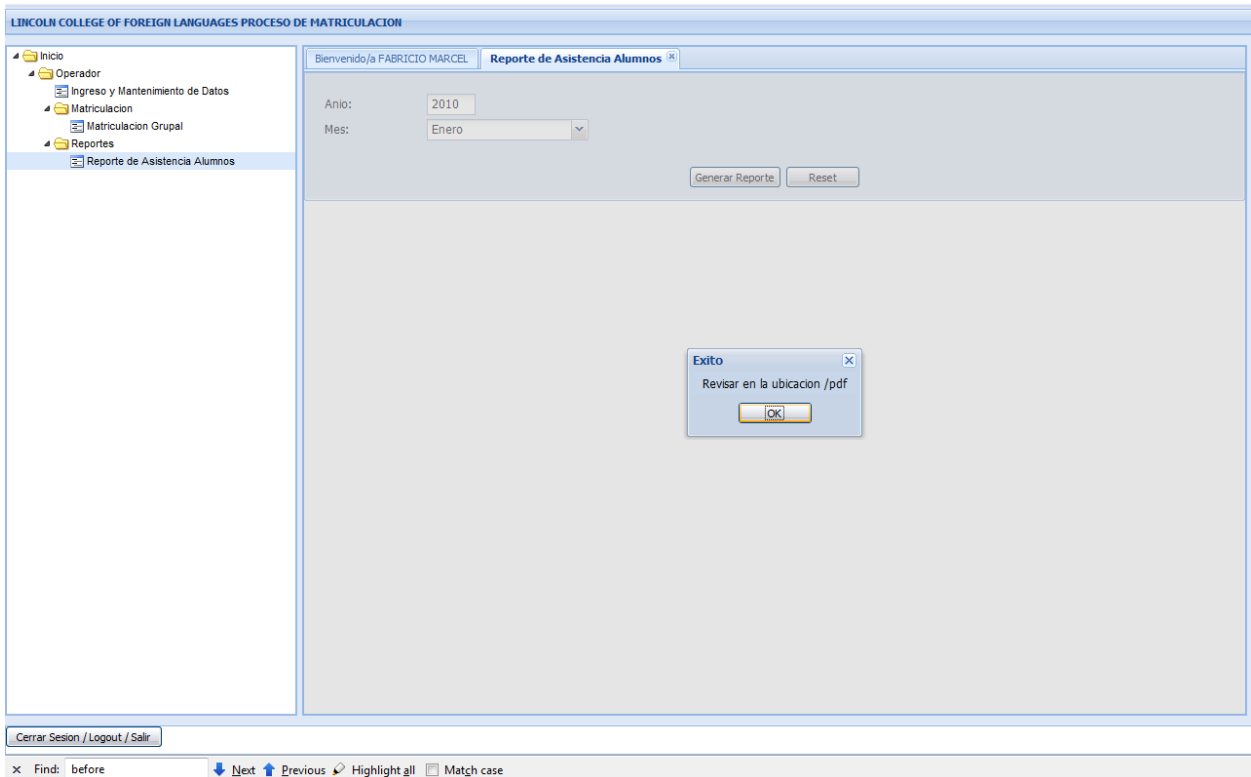
Las condiciones de ejecución del caso de prueba son que el sistema genere un reporte de asistencia para un mes en particular.

Entrada

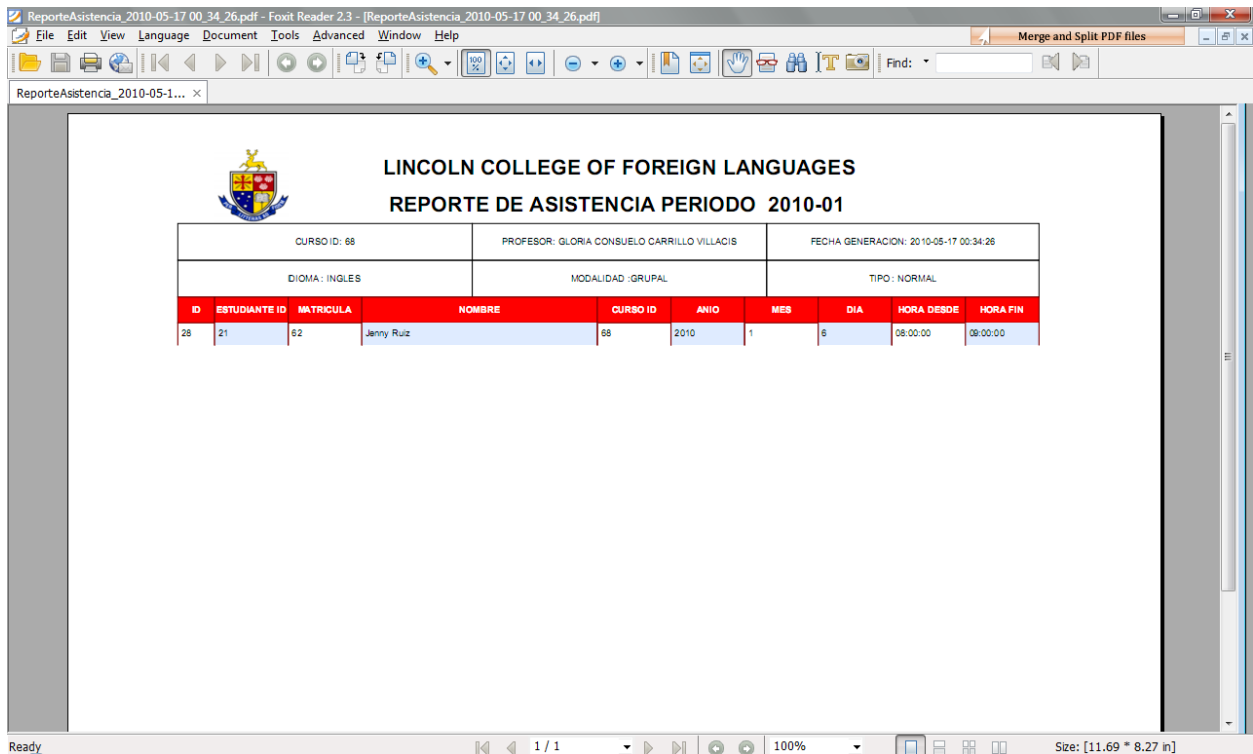
- Seleccionamos la opción “Reporte de Asistencia Alumnos” del Viewport , lo cual despliega la siguiente pantalla



- Ingresamos el año 2010 y el mes de Enero para luego presionar el botón Generar Reporte. Aparece una ventana de éxito de la creación del reporte.



- Nos vamos a la ubicación /pdf en el servidor
- Abrimos el reporte generado el día de hoy



Resultado esperado

El sistema crea un reporte de asistencia del mes y año especificados.

Evaluación de la Prueba

Prueba superada con éxito.

FASE: CONSTRUCCIÓN	ID: CNS007
NOMBRE SISTEMA:	Sistema Escolástico Lincoln College
DESCRIPCIÓN:	Especificación de Caso de Prueba: Generar Rol de Pagos de Profesores
VERSIÓN:	1.0

Descripción

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Generar Rol de Pagos de Profesores”.

Comprobar que se pueda generar un reporte de horas trabajadas calculado el salario mensual de cada profesor.

Descripción

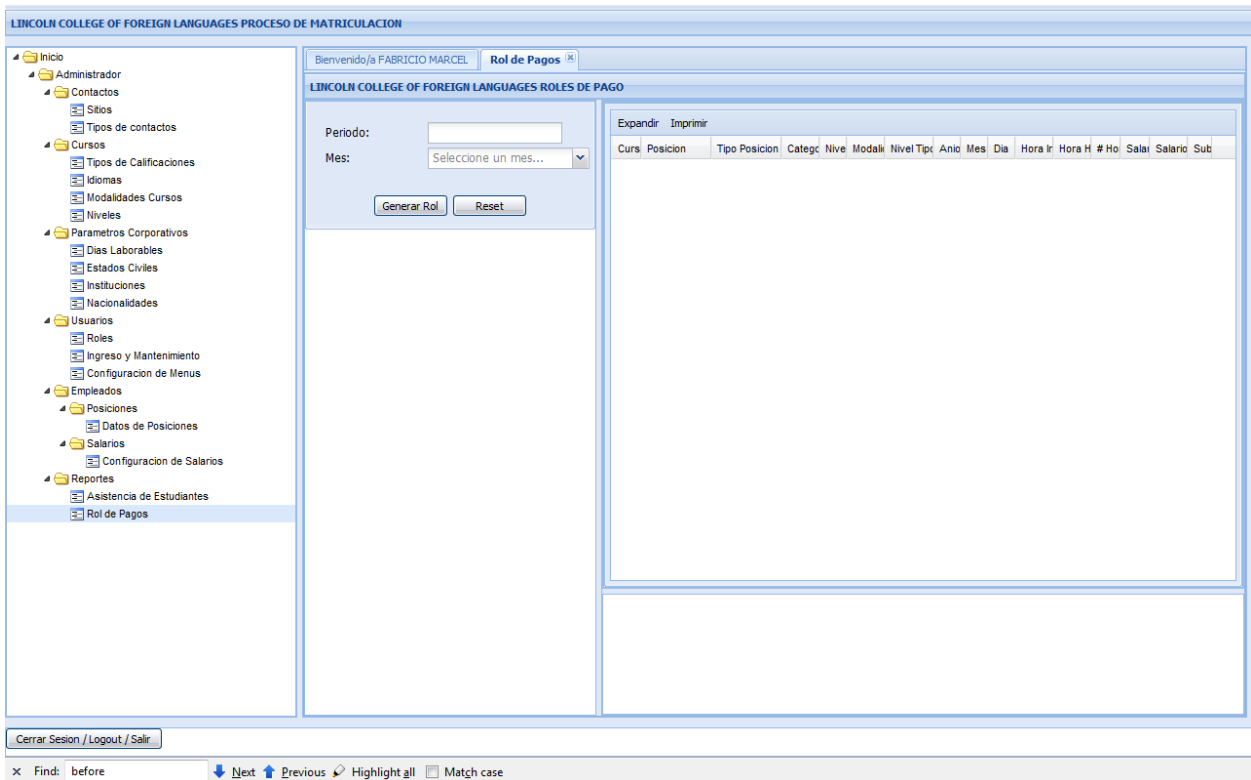
Ingresamos al sistema como Operador para generar un rol de pagos.

Condiciones de ejecución

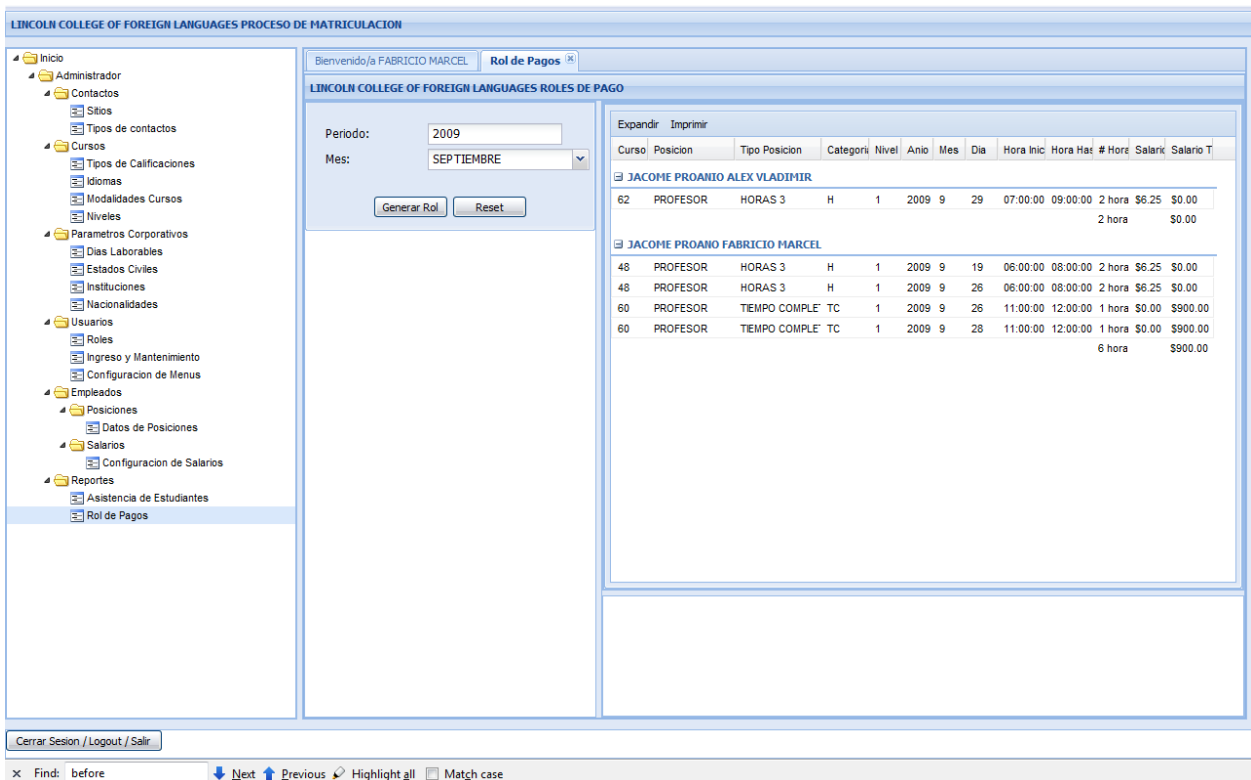
Las condiciones de ejecución del caso de prueba son que el sistema genere un rol de pagos de profesor de una forma mensual.

Entrada

- Seleccionamos la opción “Rol de Pagos” del Viewport , lo cual despliega la siguiente pantalla



- Ingresamos el año 2009 y el mes de Septiembre para luego presionar el botón Generar Rol , obtenemos la siguiente pantalla



Resultado esperado

El sistema crea un rol de pagos exitosamente.

Evaluación de la Prueba

Prueba superada con éxito

4.6 FASE DE TRANSICIÓN

Una vez concluida la fase de Construcción, se procede a capacitar a los usuarios, pero como este proyecto no va a ser puesto en producción, este permanece en el ambiente de desarrollo, sin embargo, se ha producido para esta fase el artefacto Manual de Usuario.

(ANEXO C: Manual de Usuario)

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- La metodología RUP es un proceso iterativo, incremental que permite realizar un proyecto de software de una manera planificada y ordenada, la cual deja atrás viejos paradigmas y malas prácticas. El énfasis puesto en las fases y sus disciplinas hace que se pueda producir un producto de calidad con un alto grado de escalabilidad. Este proyecto se benefició de la metodología RUP desde su concepción a través de cada una de sus fases. RUP como metodología fue creado para proyectos grandes dentro de grandes organizaciones pero se ha dado también que se puede adaptar a proyectos pequeños como el presente proyecto, donde no se tiene todos los roles ni la necesidad de realizar toda la documentación y artefactos.
- PHP más que un lenguaje de programación es una tecnología de servidor con poderosas herramientas que facilitan el trabajo a los desarrolladores. Una de las características más importantes es su capacidad de conversión inteligente entre distintos tipos de datos y su rapidez en los ambientes de producción, además de su versatilidad que dieron a este proyecto un gran apoyo.
- ActiveRecord en la capa de persistencia permitió al proyecto tomar una dirección más ordenada dentro de la arquitectura MVC que se adapta a la utilización de objetos

persistentes dejando libre al desarrollador de los accesos y operaciones a la base de datos que en ciertas aplicaciones aparecían por todos lados haciendo del mantenimiento algo imposible de realizar.

- AJAX determina una era de cambio en la utilización de las interfaces web para hacerlas asemejarse a aplicaciones tipo desktop como las programadas para los ambientes de Windows y Linux en lo que se denomina RIA (Rich Internet Applications). Una de las más importantes ventajas de AJAX es su asincronismo, el cual reparte las llamadas al servidor equitativamente y gradualmente evitando así realizar una transferencia masiva de datos que puede provocar colisiones y pérdidas de datos en la red.
- ExtJs demostró ser un poderoso framework capaz de realizar interfaces difíciles o casi imposibles con otras tecnologías como JSF , RichFaces , etc. y lo que es más importante encapsulando aspectos cruciales como la funcionalidad multibrowser , los accesos asíncronos , la transformación de objetos de servidor a objetos javascript , lo mejor de lo mejor de librerías tales como: jquery , mootools, prototype ,etc ; ofreciendo además un mecanismo de conexión con otras librerías AJAX mediante el uso de adaptadores.
- La notación JSON logra resultados excelentes por lo que no necesita ningún algoritmo de conversión para convertirse en un objeto javascript. JSON no tiene la necesidad de enviar etiquetas y datos innecesarios por lo que es una perfecta combinación con el asincronismo de AJAX.

- Se concluye que los principios del diseño que se utilizarán en un proyecto de software varían de acuerdo a la clase de aplicación a ser desarrollada. No hay un modelo general Establecido para el desarrollo del software, únicamente se definen principios que deberán ser acogidos según las necesidades y requerimientos del proyecto.
- La arquitectura MVC facilita el trabajo del equipo de software debido a la separación de responsabilidades en capas lo cual permite un mejor desarrollo, escalabilidad y estabilidad del sistema y por sobre todo un mejor mantenimiento independientemente de la tecnología a utilizarse.
- El documento de Arquitectura SAD es un artefacto primordial dentro de la fase de Elaboración, su creación es crucial para la fase de construcción. Este artefacto es evaluado por el equipo de arquitectos de datos, software y los implementadores.
Para proyectos como este con interfaces ajax se evalúa el grado de conocimientos de los usuarios del sistema y se brinda capacitación si es necesario.

RECOMENDACIONES

- Se recomienda tomar el tiempo necesario en el análisis en las etapas iniciales del desarrollo de software, ya que esto es crucial en la creación de productos de calidad dentro de los límites de tiempo y costo establecidos para un proyecto.

- Las novedades detectadas en dichas etapas o durante su evolución son causa frecuente de dificultades en el mantenimiento, baja reutilización y comportamiento defectuoso de los programas. Igualmente, las malas estimaciones realizadas al comienzo del proyecto tienen consecuencias desastrosas en cuanto a costos y plazos de entrega.
- RUP es una metodología que pone a disposición una amplia gama de artefactos, lo cual puede confundir si no se utiliza solo lo necesario. Se recomienda en proyectos pequeños no considerar todos los artefactos ya que significa una pérdida de tiempo que no es aconsejable
- El documento SAD generado en la fase de elaboración por lo que se recomienda como una guía en la fase de construcción ya que sus 4 vistas definen un bosquejo bastante detallado del sistema a construirse.
- Se recomienda mantener la ideología de un uso mezclado de accesos asíncronos y síncronos dentro de una aplicación, evitando el uso de demasiados elementos gráficos como animaciones y efectos que son potenciales distractores para los usuarios finales, lo que puede causar que se cometan errores.
- Extjs es un framework extremadamente poderoso. Desde la versión 1.0 se han realizado mejoras al core gracias a la contribución de la comunidad. La documentación no es suficiente para aclarar ciertos aspectos por lo que se debe recurrir a la técnica del trial and error, en otras palabras, aprender de las experiencias. Pero los resultados son impresionantes por lo

que se recomienda utilizar este framework que significa un desafío y una mejora en lo que tiene que ver con las interfaces de usuario. Hoy en día con la versión 3.x las posibilidades son aun más amplias. Se recomienda implementar pistas de auditoría en los procesos informáticos dentro de un sistema como el realizado en este proyecto. Estos controles deben ser correctivos, defectivos y preventivos dentro de los ámbitos lógico y físico, en especial para monitoreo de las actividades diarias de los usuarios críticos como administrador, operador y profesor.

- El desarrollo del proyecto se benefició de las tecnologías open source las cuales permitieron un desarrollo ágil en la fase de construcción.. En relación a la arquitectura, se puede destacar la versatilidad y poder de PHP en el lado del servidor; mientras que en el lado del cliente la robustez de Extjs aseguran el desarrollo de aplicaciones RIA (Rich Internet Applications) exitosas, por lo cual se recomienda su uso.

BIBLIOGRAFÍA

POLLICE GARY, Desarrollo de Software para proyectos pequeños : Un acercamiento a RUP , Addison Wesley, 2004.

AMADOR TORO y BEATRIZ JIMENEZ, Metodología para el análisis de Requisitos de Sistemas de Software, Universidad de Sevilla, 2002.

JARRET RICHARDSON y WILLIAM GWALTNEY, Guía práctica para proyectos de desarrollo exitosos, The Pragmatic Bookshelf, 2005.

HERRINGTON JACK., PHP Hacks, O'Reilly, Enero 2006.

CRANE DAVE, Ajax in Action, Manning, 2006.

KEVIN YANK, Mysql and PHP, Webmasterbase.com, 2004.

ASNDREU, R., RICART, J.; Estrategias y Sistemas de Información, Mc Graw Hill.-1998.

DAN RANSOM, Dynamic Web Forms Professional Projects, Premier Press, 2002.

DAVE TAYLOR, Creating Cool Websites with HTML, XHTML and CSS, Wiley Publishing, 2004.

MARNIE L. HUTCHESON, Software Testing Fundamentals: Methods and Metrics, John Wiley & Sons, 2003.

LUKE WELLING LAURA THOMPSON, PHP and MySQL Web Development, Sams Publishing Developer's library, 2004.

GARY POLLICE, LIZ AUGUSTINE, The Rational Unified Process Made Easy, Addison Wesley, 2004.

STAN, M., LEONARD T, Rational Unified Process For System Engineering, First Edition, 2001.

DAVID WEST, Planning a Project with the Rational Unified Process , First Edition , 2002.

TIM CONVERSE, Biblia del PHP5 y Mysql , First Edition , John Wiley and Sons , 2005.

JASON GERNER, Professional LAMP , Versión 2.2 , Wrox Publishing, 2007.

McDERMID J., Software Engineer's Reference Book, Butterworth-Heinemann Ltd, New Edition, 1992.

JESUS GARCIA, Extjs in Action, Manning Publishing; First Edition, 2010.

STEVE BLADES, Learning Extjs, Packt Publishing, First Edition, 2009.