



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN SISTEMAS EN INFORMÁTICA

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERA DE SISTEMAS**

**TEMA: DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL
SISTEMA DE CONTROL DE GRADUADOS PARA LA
UNIVERSIDAD DE LAS FUERZAS ARMADAS- ESPE,
ENMARCADO EN EL “MODELO – VISTA – CONTROLADOR
(MVC)” Y UTILIZANDO LA HERRAMIENTA “YII” COMO
FRAMEWORK DEL LENGUAJE PHP**

AUTORA: ADELA SOFÍA GALLARDO CUEVA

DIRECTOR: ING. EDGAR MONTALUISA

CODIRECTOR: ING. XIMENA LÓPEZ

LATACUNGA

2015

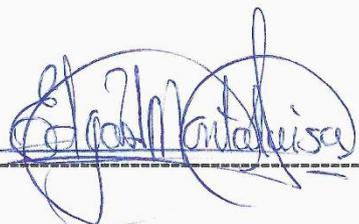


**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INENIERÍA EN SISTEMAS E INFORMÁTICA**

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DE GRADUADOS PARA LA UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE, ENMARCADO EN EL “MODELO – VISTA – CONTROLADOR (MVC)” Y UTILIZANDO LA HERRAMIENTA “YII” COMO FRAMEWORK DEL LENGUAJE PHP”** siendo parte del proyecto “Diseño de la estructura organizativa e implementación de una metodología para el desarrollo de software en la Fábrica de Software de la ESPE Extensión Latacunga.”, a fin de evaluar la factibilidad y viabilidad de la implementación de la metodología, razón por la cual se utiliza el marco teórico conceptual conjuntamente con los estudiantes Ricardo David Vásquez Garcés y Fausto Fabricio Robayo Zurita y su aplicación ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a la señora **ADELA SOFÍA GALLARDO CUEVA** para que lo sustente públicamente.

Latacunga, 20 de mayo del 2015



ING. EDGAR MONTALUISA

DIRECTOR



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SISTEMAS EN INFORMÁTICA**

AUTORÍA DE RESPONSABILIDAD

Yo, **ADELA SOFÍA GALLARDO CUEVA**, con cédula de identidad N° 0501803134, declaro que este trabajo de titulación “**DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DE GRADUADOS PARA LA UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE, ENMARCADO EN EL “MODELO – VISTA – CONTROLADOR (MVC)” Y UTILIZANDO LA HERRAMIENTA “YII” COMO FRAMEWORK DEL LENGUAJE PHP**” ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, mayo 20 del 2015

ADELA SOFÍA GALLARDO CUEVA

C.C.: 0501803134



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

AUTORIZACIÓN

Yo, **ADELA SOFÍA GALLARDO CUEVA**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **“DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DE GRADUADOS PARA LA UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE, ENMARCADO EN EL “MODELO – VISTA – CONTROLADOR (MVC)” Y UTILIZANDO LA HERRAMIENTA “YII” COMO FRAMEWORK DEL LENGUAJE PHP”** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Latacunga, mayo 20 del 2015

ADELA SOFÍA GALLARDO CUEVA

C.C.:0501803134

DEDICATORIA

Esta tesis va dedicada a mis hijos Dayis, Cami y Juanes, a mis hermanos, a mi madre, a mis sobrinas y a todas las personas que siempre estuvieron en el transcurso del desarrollo de este trabajo apoyándome en todo momento a culminar y conseguir este objetivo en mi vida.

Sofía.

AGRADECIMIENTO

Agradezco a la Universidad de las Fuerzas Armadas extensión Latacunga, por permitir ser parte del semillero de futuros profesionales, a mi Director y Codirector de Tesis por aportar con sus conocimientos y experiencia profesional que quedan impresos en éste trabajo, a toda el área de docentes que día a día imparten sus sabiduría para forjar a los futuros líderes que continuarán engrandeciendo el desarrollo del país.

Sofía.

INDICE DE CONTENIDOS

| | |
|----------------------------------|------|
| CARÁTULA..... | i |
| CERTIFICACIÓN..... | ii |
| AUTORÍA DE RESPONSABILIDAD | iii |
| AUTORIZACIÓN | iv |
| DEDICATORIA..... | v |
| AGRADECIMIENTO..... | vi |
| INDICE DE CONTENIDOS..... | vii |
| ÍNDICE DE TABLAS | x |
| ÍNDICE DE FIGURAS | xi |
| RESUMEN..... | xii |
| ABSTRACT | xiii |

CAPÍTULO I

| | |
|--|----------|
| 1. MARCO TEÓRICO | 1 |
| 1.1. Patrones de Diseño | 1 |
| 1.1.1 Patrones de creación | 3 |
| 1.1.2 Patrones estructurales | 4 |
| 1.1.3 Patrones de comportamiento | 5 |
| 1.2. Patrón de diseño “Modelo – Vista – Controlador” | 7 |
| 1.3 Arquitectura de Software | 11 |
| 1.4 Frameworks de desarrollo..... | 12 |
| 1.4.1 Tipos de Framework WEB | 13 |

| | |
|--|----|
| 1.4.2 Características de los frameworks | 13 |
| 1.5 Procesos de Desarrollo | 14 |
| 1.5.1 Codificar Y Corregir | 14 |
| 1.5.2 Modelo de Cascada | 15 |
| 1.5.3 Desarrollo Ágil | 15 |
| 1.5.4 Desarrollo Iterativo o Incremental | 15 |
| 1.5.5 Modelo en Espiral | 15 |
| 1.6 Metodología de las Tic's de la Universidad de las Fuerzas Armadas ESPE | |
| extensión Latacunga | 16 |
| 1.6.1 Fases de la Metodología | 16 |
| a. Formalización de la Necesidad | 18 |
| b. Requerimientos | 18 |
| c. Diseño | 19 |
| d. Implementación | 20 |
| e. Pruebas | 21 |
| f. Puesta en Marcha | 22 |
| g. Artefactos y Manuales | 22 |
| 1.6.2 Formalización de la Necesidad | 23 |
| 1.6.3 Requerimientos | 27 |
| 1.6.4 Diseño | 31 |
| 1.6.5 Implementación | 35 |
| 1.6.6 Pruebas | 37 |
| 1.6.7 Puesta en Marcha | 39 |
| 1.6.8 Manuales | 42 |

CAPÍTULO II

| | |
|---|-----------|
| 2 DESARROLLO DEL PRODUCTO SOFTWARE | 43 |
| 2.1 Formalización de la Necesidad | 43 |
| 2.2 Requerimientos | 43 |
| 2.3 Diseño | 43 |
| 2.4 Implementación | 43 |
| 2.5 Pruebas | 43 |
| 2.6 Liberación | 44 |

CAPÍTULO III

| | | |
|----------|---|-----------|
| 3 | MANUAL DE USUARIO Y CAPACITACIÓN AL CLIENTE..... | 45 |
|----------|---|-----------|

| | |
|-------------------------|-----------|
| CAPÍTULO IV..... | 46 |
|-------------------------|-----------|

| | | |
|----------|--|-----------|
| 4 | CONCLUSIONES Y RECOMENDACIONES..... | 46 |
|----------|--|-----------|

| | | |
|-----|-------------------|----|
| 4.1 | Conclusiones..... | 46 |
|-----|-------------------|----|

| | | |
|-----|-----------------------|----|
| 4.2 | Recomendaciones | 46 |
|-----|-----------------------|----|

| | |
|--|-----------|
| REFERENCIAS BIBLIOGRÁFICAS..... | 48 |
|--|-----------|

| | |
|---------------------|-----------|
| ANEXOS | 49 |
|---------------------|-----------|

Anexo 1. Formalización de la necesidad

Anexo 2. Requerimientos

Anexo 3. Diseño

Anexo 4. Implementación

Anexo 5. Pruebas

Anexo 6. Liberación

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Patrones de Diseño | 2 |
| Tabla 2. Patrones de diseño - Ventajas y Desventajas | 6 |
| Tabla 3. Formalización de Desarrollo de Software | 24 |
| Tabla 4. Planificación de Entregas | 26 |
| Tabla 5. Requisitos Específicos | 28 |
| Tabla 6. Diseño de Interfaces | 31 |
| Tabla 7. Diseño de base de datos | 33 |
| Tabla 8. Versión de Software | 35 |
| Tabla 9. Pruebas de Sistema | 37 |
| Tabla 10. Entrega del Sistema | 40 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1. MVC con PHP5..... | 9 |
| Figura 2. Proceso de desarrollo de software..... | 14 |
| Figura 3. Metodología para desarrollo de la Fábrica de Software | 17 |

RESUMEN

En la actualidad la organización de los institutos de educación superior establece vincular en forma estratégica las áreas académicas de la extensión y las áreas productivas del país, exigiendo de esta manera la participación del graduado en la universidad como pilar fundamental al levantamiento de información en un proceso cíclico de calidad e involucramiento con compromiso. La legislación del estado hoy en día regula como ente fundamental al graduado o graduada para la formulación del perfil profesional abalizándose en el perfil ocupacional en donde el graduado o graduada tiene la vivencia, pues tiene un papel protagónico al formar el Órgano Colegiado Académico Superior y de esta manera mantener y fortalecer la relación bilateral en cada una de los proyectos carreras del departamento de la Universidad de las Fuerzas Armadas ESPE extensión Latacunga. El producto software resultado de este proyecto permitirá tener un vínculo con el graduado en cualquier lugar contando con la información personal y profesional como datos estadísticos requeridos en cada departamento. El framework Yii y lenguaje de programación no requieren contratar licencias de uso lo que permite que a partir de la entrega del aplicativo motivo de esta tesis, permita desarrollar funcionalidades de acuerdo a nuevas necesidades sean requeridas por los departamentos en el uso del aplicativo, ya que se usa la misma tecnología para desarrollo de software que usa la Universidad, el cual permite interactuar con diferentes aplicaciones web e interactuar con el framework Yii manteniendo y respetando los procesos, procedimientos, seguridades implementados en la institución para desarrollo.

PALABRAS CLAVE:

- **INGENIERIA DE SOFTWARE**
- **SEGUIMIENTO DE EGRESADOS**
- **PLATAFORMA DIGITAL**

ABSTRACT

Today the Organization of higher education institutes sets a strategic link between the extension's academic areas and the productive areas of the country, requiring the participation of the graduate at the University as a pillar key to gather information in a cyclic process of quality and involvement within commitment. State law today regulates the graduate as a fundamental entity for the formulation of professional profile guarantying it's occupational profile where the graduate has the experience, because it has a leading role being part of an Academic Superior Collegiate Member and by this keep and strengthen the bilateral relationships in each of the projects of the Department of the Armed Forces' University extension Latacunga. The resulting software product of this project will allow having a link with the graduate anywhere, having the personal and professional information as the statistical data required in each Department. The Yii framework and programming language does not require buying licenses for use, which allows by the time of the delivery of the application, purpose of this thesis, develop functionality according to new needs that are going to be required by the departments that use this application, because it uses the same technology that is used for development of software at the University, which allows to interact with different web applications, and interact with the Yii framework, maintaining and keeping the processes, procedures, and securities implemented in the institution for development.

KEYWORDS:

- **SOFTWARE ENGINEERING**
- **MONITORING OF GRADUATES**
- **DIGITAL PLATFORM**

CAPÍTULO I

1. MARCO TEÓRICO

1.1. Patrones de Diseño

En la actividad productiva del desarrollo del Software, cada una de las tareas debe presentar su planificación para asegurar la concepción de metas, manteniendo latente dificultades y brechas que han sido solucionadas; analizando un entorno análogo podemos observar que en un edificio, un auto o un sistema informático, se debe recordar que los problemas que se presenten en el proceso alguien ya los resolvió, es ahí donde surgen los patrones de diseño los cuales ayudan a utilizar las soluciones ya probadas y documentadas por otras personas a problemas que son comunes en una situación en particular como es el caso del desarrollo del producto software.

Para los desarrolladores de sistemas, lo ideal sería contar con una colección de estos patrones con los problemas que resuelven, pero se debe considerar que los mismos tienen su grado de complejidad, partiendo del hecho que son creados por otras personas y aplicados por quienes tienen la necesidad de dar solución a un problema, sin embargo, una vez que se comprende su funcionamiento y aplicación los diseños serán mucho más flexibles, modulares y especialmente reutilizables dentro de un macro entorno que presta su facilidad de aceptación.

Gamma en su obra manifiesta que los patrones de diseño son “descriptores de clases y objetos relacionados que están particularizados para resolver un problema de diseño general en un determinado contexto” (Erich, Helm, Johnson, & Vlissides, 2003).

Existen dos criterios base para la clasificación de los patrones de Diseño como se puede observar en la Tabla 1

Tabla 1.

Patrones de Diseño

| | | Propósito | | |
|--------|--------|---|--|---|
| | | De creación | Estructurales | De comportamiento |
| Ámbito | Clase | Factory Method | Adapter (de clases) | Interpreter Template Method |
| | Objeto | Abstract Factory Builder Prototype Singleton | Adapter (de objetos) Bridge Composite Decorator Facade Flywiegth Proxy | Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor |

Fuente: (Erich, Helm, Johnson, & Vlissides, 2003)

En base al propósito; es decir, en función de lo que cada patrón realiza tenemos la siguiente clasificación: creacionales, estructurales y de comportamiento.

- **Patrones Creacionales:** Están relacionados con los procesos de creación de objetos. Sin descuidar la instancia de creación como su configuración, es decir, ayudan a encapsular y abstraer dicha creación.
- **Patrones Estructurales:** Tratan sobre la composición de clases u objetos, separan la interfaz de la implementación, se ocupan de como las clases y objetos se agrupan para formar estructuras más grandes.
- **Patrones de Comportamiento:** Describen el modo en que las clases y objetos interactúan y distribuyen sus responsabilidades, así como los algoritmos que encapsulan: es decir, muestran la comunicación entre los objetos.

Existen patrones que aplican a las clases, los mismos que se ocupan en las relaciones de clases y subclases que se crean gracias a la herencia establecida en los tiempos de compilación; así como patrones que se aplican

a los objetos y a su vez tratan de las relaciones entre ellos que pueden cambiar en tiempo de ejecución.

Las diversas formas de organizar patrones nos permiten utilizarlos juntos o en forma alterna para interactuar con sus componentes sin desvirtuar el efecto del negocio en la aplicación. “Tener muchas formas de pensar en los patrones le hará comprender mejor qué es lo que hacen, como compararlos y cuándo aplicarlos.” (Erich, Helm, Johnson, & Vlissides, 2003).

A continuación una lista resumen con los patrones de diseño orientado a objetos más utilizados según (Erich, Helm, Johnson, & Vlissides, 2003)

1.1.1 Patrones de creación

- **Abstract Factory.-** Proporciona una interfaz para crear familias de objetos o que dependen entre sí, sin especificar sus clases concretas.
- **Builder.-** Separa la construcción de un objeto complejo de su representación, de forma que el mismo proceso de construcción pueda crear diferentes representaciones.
- **Factory Method.-** Define una interfaz para crear un objeto, pero deja que sean las subclases quienes decidan qué clase instanciar.

Este patrón se utiliza cuando se tiene múltiples variantes de una sola entidad. Por ejemplo, se tiene una clase **button** la cual tiene diferentes variaciones como: `ImageButton`, `InputButton` y `FlashButton`; dependiendo del lugar, se necesita crear diferentes botones, es aquí donde se puede utilizar `Factory Method` para crear los botones.

- **Prototype.-** Especifica los tipos de objetos a crear por medio de una instancia prototípica, y crear nuevos objetos copiando este prototipo.
- **Singleton.-** Garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a ella.

Si se requiere pasar una instancia específica de una clase a otra, se puede usar `Singleton` para evitar tener que pasar la instancia vía el constructor o un argumento, con lo cual se logra acceder desde distintos lugares de la programación e incluso de diferentes clases.

1.1.2 Patrones estructurales

- **Adapter.-** Convierte la interfaz de una clase en otra distinta que es la que esperan los clientes con la finalidad de que pueda ser utilizada por una clase que de otra forma no lo haría.

Un buen ejemplo, es cuando se crea una clase dominio para las clases de tabla y llamar sus funciones una a una, se puede encapsular todos estos métodos en un método que utilice una clase adaptador. Esto no sólo permite reutilizar cualquier acción que se desee, sino también evitará que se tenga que reescribir el código si se requiere usar la misma acción en un sitio distinto.

- **Bridge.-** Desvincula una abstracción de su implementación, de manera que ambas puedan variar de forma independiente.
- **Composite.-** Combina objetos en estructuras de árbol para representar jerarquías de parte-todo, esto permite que los clientes traten de manera uniforme a los objetos individuales y a los compuestos.
- **Decorator.-** Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad.

El momento más adecuado para usar Decorator es cuando se tiene una entidad que necesita tener un nuevo comportamiento sólo si la situación así lo requiere, por ejemplo, se tiene un enlace HTML que realiza cosas diferentes dependiendo de la página en la que se encuentre, como puede ser mostrar el link subrayado si se encuentra en una página y resaltado si está en otra.

- **Facade.-** Proporciona una interfaz unificada para un conjunto de interfaces de un subsistema
- **Flyweight.-** Usa el compartimiento para permitir un gran número de objetos de grano fino de forma eficiente.
- **Proxy.-** Proporciona un sustituto o representante de otro objeto para controlar el acceso a éste.

1.1.3 Patrones de comportamiento

- **Chain of Responsibility.-** Evita acoplar el emisor de una petición a su receptor, al dar a más de un objeto la posibilidad de responder a la petición, para esto crea una cadena con los objetos receptores y pasa la petición a través de la cadena hasta que esta sea tratada por algún objeto.
- **Command.-** Encapsula una petición en un objeto, permitiendo así paramétrica a los clientes con distintas peticiones, encolar o llevar un registro de las peticiones y poder deshacer la operaciones.
- **Interpreter.-** Dado un lenguaje, define una representación de su gramática junto con un intérprete que usa dicha representación para interpretar las sentencias del lenguaje.
- **Iterator.-** Proporciona un modo de acceder secuencialmente a los elementos de un objeto agregado sin exponer su representación interna.
- **Mediator.-** Define un objeto que encapsula cómo interactúan un conjunto de objetos. Promueve un bajo acoplamiento al evitar que los objetos se refieran unos a otros explícitamente, y permite variar la interacción entre ellos de forma independiente.
- **Memento.-** Representa y externaliza el estado interno de un objeto sin violar la encapsulación, de forma que éste puede volver a dicho estado más tarde.
- **Observer.-** Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambia de estado se notifica y actualizan automáticamente todos los objetos.
- **State.-** Permite que un objeto modifique su comportamiento cada vez que cambia su estado interno. Parecerá que cambia la clase del objeto.
- **Strategy.-** Define una familia de algoritmos, encapsula uno de ellos y los hace intercambiables. Permite que un algoritmo varíe independientemente de los clientes que lo usan.
- **Template Method.-** Define en una operación el esqueleto de un algoritmo, delegando en las subclases algunos de sus pasos. Permite

que las subclases redefinan ciertos pasos del algoritmo sin cambiar su estructura.

- **Visitor.-** Representa una operación sobre los elementos de una estructura de objetos. Permite definir una nueva operación sin cambiar las clases de los elementos sobre los que opera.

El uso de patrones de diseño no es solo un estilo arquitectural es también una forma de afrontar los proyectos a nivel de trabajo del equipo de desarrollo, ciclo de vida del proyecto, lenguaje a utilizar con los expertos en el negocio, etc.; sin embargo también identifica una serie de patrones de diseño y estilo de arquitectura concreta, es también, una aproximación concreta para diseñar software basándose sobre todo en la importancia del dominio del negocio, sus elementos y comportamientos y las relaciones entre ellos; tiene sus ventajas y desventajas las cuales se resumen en la Tabla 2

Tabla 2.

Patrones de diseño - Ventajas y Desventajas

| Ventajas | Desventajas |
|--|---|
| Permite reducir operaciones complejas presentando son soluciones concretas: Los patrones son recetas de diseño, se pueden clasificar, manteniendo independencia para exponerlos como simple. | No posee ingeniería inversa |
| Las soluciones técnicas dada una determinada situación presenta un escenario interesante de desacoplamiento entre los objetos utilizando un lenguaje específico y otros | Es difícil interpretar el patrón: los patrones no se reflejan en el código empleado. |
| Se aplican en soluciones muy comunes: Proceden de la experiencia y tiene utilidad demostrada en problemas del diseño orientado a objetos. | El patrón de diseño es sumamente costoso y difícil en lenguajes que no se alinean al paradigma de desarrollo. |
| Presenta soluciones simples: Resuelven un problema particular en la mayoría de aplicaciones utilizando un pequeño número de clases con flexibilidad y finamente con calidad de software. | Su utilización es complicada por el hecho que las tecnologías actuales incentivan a los desarrolladores para particionar la aplicación en la fase de diseño |



| | |
|--|---|
| La utilidad se vuelve cada vez más importante: Nos permite reducir tiempo de desarrollo por lo tanto costo incrementando calidad | Tiempo: El tiempo dedicación en la fase inicial es elevado siendo relativo ya que su ventaja de mantenimiento es productiva |
| Estructura general: Su estructura es conocida por todos los programadores, de tal manera que su producto al ser independiente no resulte distinta de su forma de trabajar. | |
| Permiten tener una estructura de código común a todos los proyectos que implemente una funcionalidad genérica. | |
| Una aplicación con MVC es mucho más mantenible, extensible y modificable | |

En la Tabla 2 se puede evidenciar que las ventajas da una diferencia competitiva en el uso de patrones de diseño, considerando que existe trabajo ya realizado, probado y aceptado entorno de desarrollo de producto software aportando con una solución satisfactoria en tiempo y costo aceptable sobre todo con una calidad marcada.

En diseño de aplicaciones es fundamental anticiparse a nuevos requisitos y a cambios en los ya existentes de manera que el diseño soporte estas variaciones y evolucionen sin mayores complicaciones, sin correr el riesgo de rediseñar por completo. Los cambios no previstos pueden implicar el rediseño de la aplicación lo que involucra: redefiniciones y nuevas implementaciones de clases, modificar los clientes y repetir el ciclo de pruebas, lo cual siempre resulta ser muy costoso. Los patrones de diseño ayudan a evitar este tipo de problemas al dejar que algún aspecto de la estructura del sistema varíe de manera independiente, lo que da como resultado sistemas más robustos frente a cambios concretos.

1.2. Patrón de diseño “Modelo – Vista – Controlador”

Para el diseño de aplicaciones Web con interfaces enriquecidas y que ofrezcan una fuerte interactividad con el usuario se utiliza el patrón de diseño Modelo-Vista-Controlador. Aquí se conjuga la interfaz lógica y la lógica de negocio esto se debe a la abstracción del objeto en un orden de relación

jerárquica entre ellos, es decir la herencia guardan un relación de orden jerárquico entre si donde se pueden establecer claras diferencias prácticas, esto trae consigo mayor trabajo y una alta probabilidad de error.

Con este patrón se trata de realizar un diseño que separe la vista del modelo, con la finalidad de mejorar la reusabilidad, de esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

En el patrón de diseño de software Modelo-Vista-Controlador (MVC) todo el proceso está dividido en 3 capas:

- **Modelo:** En esta capa se hace el levantamiento de todos los objetos que debe utilizar el sistema, es decir, es el proveedor de recursos. Es en donde se encapsulan los datos y la lógica del negocio, esto hace que sea independiente de la base de datos que se utilice así como de los medios de representación de los datos.
- **Vista:** Es la capa de presentación la cual muestra la información del modelo al usuario, se agrupan todas las clases y archivos que tengan relación de la interfaz de usuario. Cada vista tiene asociado un componente controlador que se indica a continuación.
- **Controlador:** El controlador viene a ser el orquestador de los diferentes eventos generados en la interfaz de usuario, se encarga de llamar en el modelo al experto del negocio que sabe que es lo que hay que hacer con la petición del usuario. Una vez que el modelo ha realizado su tarea se lo comunica al controlador. El controlador invoca a la vista o interfaz para que se actualice con los cambios hechos en el modelo. En resumen, el Controlador es el que escucha los cambios en la vista y se los envía al modelo, el cual le regresa los datos a la vista, es un ciclo donde cada acción del usuario causa que se inicie un nuevo ciclo.

En la Figura a continuación muestran las responsabilidades de cada capa del patrón de diseño MVC y cómo interactúan entre sí.

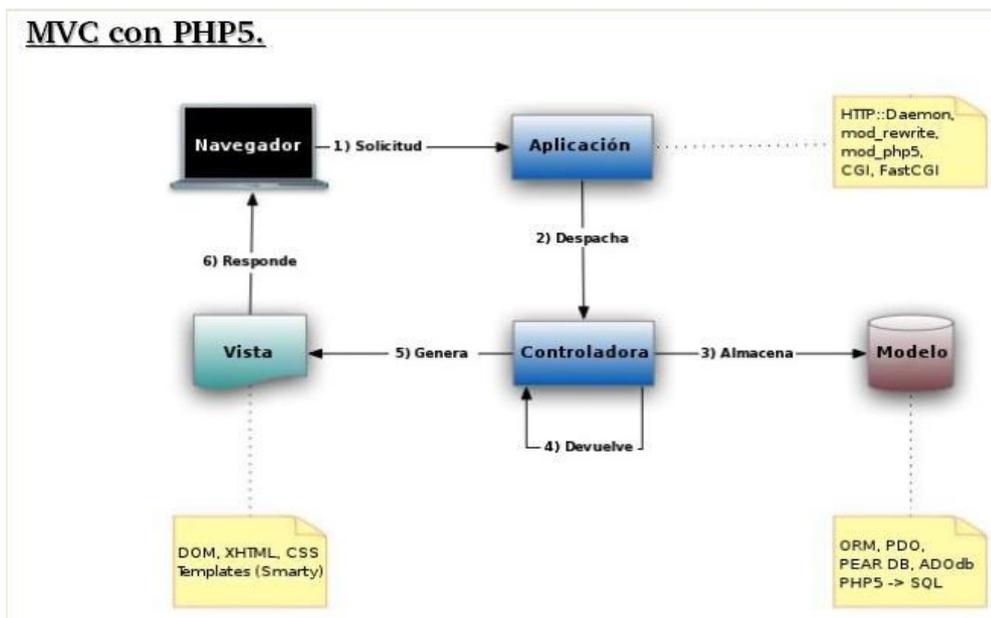


Figura 1. MVC con PHP5

Fuente: (Caballero & Lara, 2006)

La interacción que se muestra en la Figura 1 es la siguiente:

1. Solicitud: El usuario ejecuta un navegador, pide un URL solicitando a la Aplicación.
2. Despacha: La Aplicación delega a la capa **Controladora** la petición del usuario.
3. Almacena: Consulta en la capa **Modelo** por medio de la capa de datos del manejador de base de datos existente.
4. Devuelve: La capa de Datos devuelve los resultados en datos puros a la capa **Controladora**.
5. Genera: La capa **Controladora** genera la capa de **Vista** en base a los tipos de datos generados.
6. Responde: La **Vista** generada se envía como respuesta a la solicitud del navegador.

Con lo expuesto se resume las responsabilidades básicas de cada componente del patrón MVC de la siguiente manera:

- El **modelo** es responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio.
- El **controlador** es responsable de:
 - Recibe los eventos de entrada.
 - En base a reglas de gestión de eventos se realizan peticiones al modelo o a las vistas.
- Las **vistas** son responsables de:
 - Recibir datos del modelo y los muestran al usuario.
 - Tienen un registro de su controlador asociado (normalmente porque además lo instancia).

MVC implementa varios patrones a la vez, así tenemos que la relación entre la Vista y el Controlador es una implementación del patrón Strategy. MVC utiliza otros patrones de diseño, tales como el Factory Method y el Decorator. Factory Method se utiliza para especificar la clase controlador predeterminada de una vista y Decorator para añadir la capacidad de desplazamiento de una vista, sin embargo, las principales relaciones se dan entre los patrones Observer, Composite y Strategy.

“Si una aplicación no está diseñada siguiendo el patrón de diseño MVC, inevitablemente se tendrá problemas de escalabilidad, mantenimiento y de extensión” (Caballero & Lara, 2006). Se puede afirmar que en aplicaciones Web la programación utilizando el patrón de diseño MVC no es una alternativa sino una necesidad que evitará en el futuro muchos dolores de cabeza gracias a sus características que en resumen son las siguientes:

- Patrón de diseño orientado a objetos.
- Adecuado para aplicaciones Web con alta interacción humana. Separa clara y consistentemente las preocupaciones en las capas indicadas anteriormente.
- Permite múltiples representaciones (vistas) de la misma información
- (modelo).
- Facilidad para agregar, eliminar o modificar interfaces de usuario. Facilita los desarrollos simultáneos con actualizaciones de interfaces,

lógica del negocio o incluso introducir una nueva aplicación sin la necesidad de afectar a otro código fuente.

- Garantiza la reducción del código fundamentalmente porque los modelos pueden ser utilizados en varias vistas.
- Los desarrolladores centran su atención en un solo aspecto de la aplicación al mismo tiempo.

Estas características conducen a que un proyecto de software Web se implemente de una forma ágil con un producto sostenible y mantenible en el tiempo y que en el futuro puede migrar en cualquier dirección. Estas son las razones fundamentales por las que se utilizará el patrón MVC en el presente proyecto.

1.3 Arquitectura de Software

El diseño global del sistema usando un patrón definido se denomina arquitectura.

La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad y disponibilidad. (Kruchten, 1995)

La arquitectura de software de un programa o de un sistema computacional está definida por la estructura, comprendida por los elementos de software, las propiedades visibles de esos elementos y las relaciones entre ellos. (Gomaa, 2011)

Incluyendo:

- La descripción de los componentes con los cuales se construyen los sistemas
- Las interacciones entre esos componentes
- Patrones para guiar la composición
- Restricciones sobre dichos patrones

- Componentes: servidores, clientes, bases de datos, filtros, capas en un sistema jerárquico, etc.
- Interacciones: llamadas a procedimientos, protocolos C/S, protocolos de acceso a BD, etc.

La arquitectura de software se encarga de:

- Diseño preliminar o de alto nivel.
- Organización a alto nivel del sistema, incluyendo aspectos como la descripción y análisis de propiedades relativas a su estructura y control global, los protocolos de comunicación y sincronización utilizados, la distribución física del sistema y sus componentes, etc.
- Otros aspectos relacionados con el desarrollo del sistema y su evolución y adaptación al cambio:

Composición, reconfiguración, reutilización, escalabilidad, mantenibilidad, etc.

Los aspectos que la arquitectura de software no se ocupa:

- Diseño detallado.
- Diseño de algoritmos.
- Diseño de estructuras de datos.

1.4 Frameworks de desarrollo

La palabra Framework por su conceptualización es utilizada en diversas áreas de negocio, académica, investigación, etc. Por ejemplo encontramos frameworks para el fortalecimiento en estimulación temprana, en el área legislativa en contratación pública, así como también en el área de software no tiene una utilización específica, en este trabajo teórico práctico lo orientamos al desarrollo de software y en concreto en el ámbito de aplicaciones web; ya que como objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un framework Web, por tanto, podemos definirlo como un conjunto de partes específicas que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web sin llegar a obtener los límites tope del mismo,

permitiendo así su crecimiento y complejidad a lineado a un esquema y misión empresarial.

1.4.1 Tipos de Framework WEB

“Existen varios tipos de frameworks Web: orientados a la interfaz de usuario, como Java Server Faces, orientados a aplicaciones de publicación de documentos, como Coocon, orientados a la parte de control de eventos, como Struts y algunos que incluyen varios elementos como Tapestry.

La mayoría de frameworks Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC o con el modelo 2 de Servlets y JSP, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación.” (Gutiérrez., 2006)

1.4.2 Características de los frameworks

A continuación enunciamos una serie de características que podemos encontrar en prácticamente todos los frameworks existentes.

Tabla 3.
Características de los Frameworks

| | |
|--|--|
| Abstracción de URLs y sesiones. | No es necesario manipular directamente las URLs ni las sesiones, el framework ya se encarga de hacerlo. |
| Acceso a datos | Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en Base de Datos, XML, etc. |
| Controladores. | La mayoría de frameworks implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto. |
| Autenticación y control Incluyen mecanismos para la identificación de usuarios de acceso. | Incluyen mecanismos para la identificación de usuarios mediante login y password y permiten restringir el acceso a determinadas páginas a determinados usuarios. |
| Internacionalización. | |
| Separación entre diseño y contenido. | |

Fuente: (Gutiérrez., 2006)

1.5 Procesos de Desarrollo

Un proceso de desarrollo de software o llamado ciclo de vida es la estructura básica para el desarrollo del producto software, existes varios modelos a seguir para establecer el proceso de desarrollo de software de manera que éste cumpla los requisitos del cliente tanto en tiempo, costo, alcance y calidad. Este proceso es afectado a las habilidades e ingenio del grupo de trabajo.

Lo indicado en términos globales se muestra en la Figura 3



Figura 2. Proceso de desarrollo de software

Fuente: (Ivar, 2004)

El proceso de desarrollo de software no es único. Existen varios modelos para el para el desarrollo muchos cuentan con pros y contras, en el proyecto se debería escoger el que más se ajuste a la necesidad o también se puede trabajar con modelos híbridos, es decir combinar varios modelos. Existen muchos modelos genéricos de proceso de software que vamos a revisar brevemente a continuación.

1.5.1 Codificar Y Corregir

El ciclo de vida codificación y corrección es más una estrategia a la falta de experiencia o presión que se ejerce sobre el grupo de desarrollo para cumplir con el tiempo de entrega, sin dedicar tiempo al diseño, el grupo de trabajo empieza a trabajar directamente en el código y realiza sus pruebas, obteniendo inevitables errores que se solucionan antes de la entrega del producto Software.

1.5.2 Modelo de Cascada

El modelo de cascada es un proceso donde el grupo de trabajo tiene que seguir las siguientes fases sucesivamente:

- 1) Especificación de requisitos
- 2) Diseño de software
- 3) Integración
- 4) Pruebas(o validación)
- 5) Despliegue(o instalación)
- 6) Mantenimiento

Para pasar de fase se debe finalizar la fase anterior, se puede hacer una revisión al terminar la fase para estar seguro o poder corregir y seguir a la siguiente fase.

1.5.3 Desarrollo Ágil

“El desarrollo ágil de software utiliza un desarrollo iterativo como base para abogar por un punto de vista más ligero y más centrado en las personas que en el caso de las soluciones tradicionales. Los procesos ágiles utilizan retroalimentación en lugar de planificación, como principal mecanismo de control. La retroalimentación se canaliza por medio de pruebas periódicas y frecuentes versiones del software.” (Mann, 1997)

1.5.4 Desarrollo Iterativo o Incremental

El desarrollo iterativo recomienda la construcción de secciones reducidas del software que poco a poco se irán incrementando para facilitar la detención de errores antes de que sea demasiado tarde, el desarrollo interactivo es ideal para clientes que no tienen una idea clara de lo que desean.

1.5.5 Modelo en Espiral

La principal característica del modelo en espiral es la gestión de riesgos de forma periódica en el ciclo de desarrollo. La ventaja de este modelo es que tiene un análisis interactivo y concienzudo de los riesgos que nos sirve especialmente en sistemas de gran escala. El espiral se visualiza como un proceso que pasa a través de algunas interacciones con el diagrama de los cuatro cuadrantes:

- 1) Crea planes para identificar el objetivo

- 2) Analiza los riesgos es la fase principal del modelo
- 3) Desarrollo y Verificación
- 4) Planificación de Software

Cada proyecto de software requiere de una forma particular para abordar el problema. Las propuestas comerciales y académicas actuales promueven procesos iterativos.

1.6 Metodología de las Tic's de la Universidad de las Fuerzas Armadas ESPE extensión Latacunga

La metodología de desarrollo de software constituye la guía para el proceso de desarrollo de software, debiendo acoplarse a los procesos internos de las empresas de desarrollo. La metodología deberá proporcionar las guías necesarias que orienten el desarrollo, proporcionando los suministros necesarios para documentar el trabajo realizado sin llegar a constituirse en una receta que conlleve la creación de documentos que no aportan significativamente a la obtención del producto software.

En la ESPE Extensión Latacunga se vive una realidad particular, producto del régimen militar que guía los procesos administrativos que están orientados hacia los ejes de Educación, Investigación y Extensión. En este ámbito se propone una metodología que sea capaz de ofrecer un camino seguro que guíe el desarrollo de software permitiendo lograr un producto software que cumpla los requerimientos internos de la Universidad así como proyectos externos. Estos proyectos deberán adaptarse para poder ser desarrollados en un entorno regido por las políticas institucionales.

La Metodología proporcionará un conjunto de fases a ser seguidas y dentro de las cuales se desarrollarán un grupo específico de entregables que sustentarán el trabajo realizado, propiciando la entrega de productos de software y la consecución de un buen nivel de calidad reflejado en la operatividad de la aplicación.

1.6.1 Fases de la Metodología

La metodología de desarrollo software que es objeto de esta propuesta consta de varias fases que orientarán el desarrollo del producto software en un proceso que se ejecuta al iniciar el proyecto que no deberá exceder de 3

días, un proceso iterativo el cual dará como resultado el desarrollo de versiones de software con operatividad autónoma que se ejecutarán en un tiempo de 2 a 4 semanas, y un proceso de entrega donde el usuario final podrá usar el sistema para ejecutar sus procesos administrativos y se proporcionará la capacitación en un tiempo entre 3 a 5 días.

Las fases de la Metodología de Desarrollo de Software propuesta que plantea, se esquematizan en la Figura 1.3 y son las siguientes:

- Formalización de la Necesidad
- Requerimientos
- Diseño
- Implementación
- Pruebas
- Puesta en Marcha

La metodología propicia el desarrollo en un entorno en el cual los requerimientos se presentan de forma urgente y en la cual el contacto con el dueño del procesos suele ser dificultoso por cuestiones de la organización y los procesos internos, así como la urgencia de contar con la herramienta de software que ayude a acelerar el trabajo de forma automatizada.



Figura 3. Metodología para desarrollo de la Fábrica de Software

Fuente: (López & Montaluisa, 2015)

a. Formalización de la Necesidad

La fase de Formalización de la Necesidad constituye el primer proceso de la metodología que se ejecutará por única vez al inicio del proceso metodológico.

Esta fase determina el inicio del proyecto, propiciada por la iniciativa de desarrollo de una aplicación de software como respuesta a una necesidad. Considerando las circunstancias que rodean el entorno en el cual se originan estas necesidades, en esta fase se establece el inicio del proyecto de desarrollo mediante una disposición que es propiciada por la Unidad Operativa Requirente, esta necesidad se modelará mediante un pedido formal que se denominará "Formalización de la Necesidad". La Formalización de la Necesidad establecerá los responsables y una primera indagación de lo requerido, citando las leyes, reglamentos y procesos establecidos que respalden dicha necesidad.

La Formalización de la Necesidad tendrá como finalidad dar inicio formalmente al proceso de desarrollo de software, comprometer a los actores a aportar en todas las fases requeridas para el desarrollo, e identificar el entorno en el cual se desarrollará y se ejecutará la aplicación de software desarrollada.

En esta fase se establecerá un cronograma de entregas parciales que determinará el alcance del proyecto y de las versiones de software a desarrollarse, estas versiones tendrán autonomía y que en su conjunto conformarán la aplicación de software resultante del proyecto.

Se propicia garantizar los compromisos adquiridos tanto por parte del desarrollador como del requirente.

b. Requerimientos

La fase de Requerimientos es la segunda fase del proceso metodológico, y se constituye como un componente que es parte del proceso repetitivo para el desarrollo de las versiones de software. Esta fase se ejecutará cada vez que se repita el proceso de desarrollo de una nueva versión, en esta fase se detallará la descripción formal de los requerimientos logrando plasmar el alcance del sistema.

Se considera en la metodología que el desarrollo del proyecto se ejecuta en un entorno complejo que dificulta la obtención de los requerimientos y debido al corto tiempo que se establece para el desarrollo de los proyectos se propone la esquematización de historias de usuarios con la variante que el técnico responsable de la elaboración del sistema será el encargado de recopilar y plasmar en el formato la información necesaria para detallar el requerimiento y luego deberá confrontarlo con el usuario responsable del proceso a fin de legalizarlo. Para esta fase deberá aprovechar al máximo el tiempo disponible del personal operativo responsable de los procesos y recopilar la información necesaria aplicando técnicas como:

- Entrevistas al personal involucrado con los procesos a automatizar
- Recopilación de documentos de apoyo y formatos que hayan sido establecidos para el trabajo diario
- Observación de reglamentos e instructivos desarrollados para la ejecución de proyectos que están relacionados con el requerimiento

Los requerimientos describirán de la mejor forma posible la información necesaria para obtener versiones de software de rápida entrega y que contemplen funcionalidades que el usuario final pueda iniciar a ejecutar, los requerimientos serán agrupados y levantados desde dos puntos de vista:

- La necesidad del usuario
- El criterio del técnico que permitirá estructurar paquetes de software de uso autónomo.

En la definición de los requerimientos se establecerá una sección que permita establecer los aspectos necesarios que serán validados en las siguientes fases del desarrollo a fin de establecer si el software desarrollado cumple las expectativas del usuario que lo requiere.

c. Diseño

La fase de diseño constituye la tercera fase y es parte del proceso iterativo de desarrollo de la aplicación de software, su ejecución se refleja en los modelos producto de los requerimientos establecidos para la versión de software en ejecución.

El proceso de diseño contemplará la esquematización de los requerimientos agrupados por funcionalidades, por consiguiente generará modelos parciales que provean funcionalidad independientes.

La fase de diseño considerará los siguientes aspectos que deberán ser modelados:

- **Diseño de Datos.-** El diseño establecerá el modelo de datos que dará soporte al sistema en desarrollo, se realizará mediante la aplicación de software relacionado directamente con el manejador de base de datos que se haya definido para el proyecto. Por ejemplo para MySQL se usará MySQL Workbench.
- **Diseño de Interfaces.-** El Diseño de interfaces se establecerá de acuerdo al Framework a utilizarse, se realizará un único diseño base para todas las interfaces de mantenimiento de las tablas en la cual se establecerá el posicionamiento para:
 - Texto en el formulario
 - Cajas de entrada de datos (texto, combos, casillas de selección, etc.)
 - Botones de comando y enlaces de funcionalidad
 - Gráficos y logos

Se personalizará aquellas interfaces que requieran de complejidad en la implementación y que difieren del esquema general de entradas y presentación de datos. El diseño de interfaces determina en forma gráfica mediante esquemas la distribución y operatividad deseada de la interface la que concordará con los requerimientos establecidos.

d. Implementación

La fase de Implementación es la cuarta fase del proceso iterativo de desarrollo y se enfocará en obtener código de forma rápida, con la mayor fiabilidad posible y con un alto grado de calidad.

Para lograr un estilo homogéneo y eficiente se deberá observar los diseños establecidos de interfaces y se utilizará técnicas de programación que permitan acelerar la generación de código, tal como Frameworks que

posibiliten generar código fuente con una arquitectura de software MVC (Modelo Vista Controlador).

El objetivo al integrar el Framework y la arquitectura MVC es permitir un desarrollo rápido de cada módulo y las versiones establecidas en el cronograma de entregas, pasando de la necesidad al producto software de la forma más rápida posible y la entrega al usuario final luego de las pruebas y validaciones necesarias.

La Implementación generará versiones de software que serán autónomos y fácilmente acoplables en un producto final.

El proceso para la generación de código establece seguir varios pasos de forma metódica, los que tomando como insumo los diseños de datos e interfaces generarán el código:

- Creación de Tablas de Datos en el manejador de base de datos
- Uso del generador de código
- Personalización del Modelo
- Personalización de la Vista
- Personalización del Controlador
- Pruebas Unitarias (Funcionalidad del código)

Para la implementación, así como para el resto de fases se recomienda el uso de herramientas de software dentro del marco que regula a las instituciones públicas. El decreto 1014 recomienda el uso del Software Libre, y en este marco podemos usar herramientas de desarrollo como NetBeans, así como herramientas de gestión de versiones como SubVersion (SVN).

e. Pruebas

La fase Pruebas es la quinta fase y es parte del proceso iterativo de desarrollo. Una vez finalizada la generación del código se realizarán pruebas del software a fin de garantizar la calidad mediante la concordancia entre los requerimientos, el diseño y el producto resultante.

Las pruebas se establecerán en 2 puntos a fin de minimizar los fallos residuales en el software:

- **Pruebas de Desarrollo.-** Las realizará el técnico a cargo del desarrollo y se basa en examinar de forma detallada los objetos creados y su funcionalidad. Propicia la eliminación de errores a nivel de código la funcionalidad individual del módulo.
- **Pruebas de Función.-** Las realizará un técnico a fin encontrar residuos de fallos que no han sido detectados y también establecerá la correcta relación entre los módulos desde el punto de vista funcional del software, de esta manera se asegura que la versión de software sea un elemento entregable al usuario final. En esta evaluación se contrapondrá los requerimientos establecidos para la versión de software contra el desempeño de la aplicación desarrollada.

Conseguiremos entonces una versión que pasará a estado de producción y podrá ser implantada.

f. Puesta en Marcha

La Fase de puesta en marcha se ejecutará al finalizar cada una de las versiones de software planificadas. La ejecución de esta fase requiere la puesta en marcha de la versión de software, realizando paralelamente la capacitación sobre las funcionalidades incluidas en la determinada versión.

Se orientará al usuario final a utilizar las funcionalidades implementadas y se determinará el grado de satisfacción sobre los requerimientos.

Se realizará un Acta de Conformidad en la cual se indicará las funcionalidades implementadas en la versión y se registrarán las novedades suscitadas en la capacitación.

Mientras se avanza en el proceso de Puesta en Marcha por parte de un técnico, el grupo de trabajo iniciará inmediatamente a trabajar en la nueva versión.

Los fallos residuales detectados en la Puesta en Marcha serán incluidos como parte de la nueva versión a fin de viabilizar de forma fluida el desarrollo del software.

g. Artefactos y Manuales

Para lograr la operatividad de la metodología se plantea un conjunto de artefactos de apoyo, que permitirán documentar el proceso de desarrollo e

integrar el esfuerzo de las personas involucradas en el proyecto. Estos artefactos se establecerán en cada una de las fases de la metodología.

Se requiere además de los formatos de base para los manuales que estarán a disposición del cliente o usuario final para poder implantar el sistema o capacitar a nuevos funcionarios que no fueron parte del grupo de personal capacitado durante la ejecución del proyecto.

Para poder organizar los entregables que se definen en la metodología se ha establecido un árbol de directorios que contendrán cada uno de los entregables de forma organizada y nos permitirán que todos los miembros del equipo de desarrollo sean capaces de encontrar fácilmente los artefactos que han desarrollado otros colaboradores.

- Raíz [Repositorio de Proyectos de Desarrollo]
 - Proyecto [Nombre Corto del Proyecto]
 - Entregables [Documentos de Texto y Modelos]
 - Código [Versiones de Código Fuente]
 - Sistema Versión 1.0 [Primera Versión]
 - Sistema Versión 2.0 [Segunda Versión]
 - Manuales [Versiones de Manuales]

Una vez establecido el repositorio para los artefactos que serán desarrollados como producto de la metodología procederemos a detallar cada uno de ellos de acuerdo a la fase en la cual se elaborarán.

1.6.2. Formalización de la Necesidad

En esta primera fase se establecen los documentos que permiten dar legalidad al pedido y proporcionan el marco de responsabilidad de las áreas involucradas en el futuro desarrollo.

Se plantea la utilización de los siguientes documentos:

- **Formalización de Desarrollo de Software.**- Este documento recopila la necesidad del desarrollo expresado en una disposición, memorando o pedido formal para el desarrollo de la aplicación. Estará compuesto por una descripción general del proyecto de desarrollo, la especificación de los responsables para la ejecución del proyecto,

legalizado por las firmas del requirente y del responsable del grupo de desarrollo.

Tabla 3.

Formalización de Desarrollo de Software

| | | |
|--|-------------------|--|
| Plantilla: | | |
| “FORMALIZACIÓN DE DESARROLLO DE SOFTWARE” | | |
| Función de la Plantilla: | | |
| <p>Formalizar el pedido de desarrollo de software, logrando la identificación de las entidades participantes en el proceso de desarrollo, y realizando acuerdos que permitirán una adecuada planificación y garanticen la participación activa tanto del grupo de desarrollo como del requirente.</p> | | |
| Objetivos de la Plantilla: | | |
| <ul style="list-style-type: none"> • Formalizar la necesidad presentada mediante un pedido verbal o mediante una disposición documentada. • Identificar las personas que participarán en el proceso de desarrollo tanto de técnicos como de funcionarios requirentes. • Comprometer la participación del grupo de desarrollo y del requirente mediante el suministro de información y la entrega en los plazos establecidos del producto de software. | | |
| Alcance de la Plantilla: | | |
| <p>Esta plantilla se usará al inicio del proyecto para formalizar el pedido.</p> | | |
| Notas: | | |
| <p>Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) y su nombre seguirá la nomenclatura siguiente: Formalizacion.[nombre_de_proyecto].v[número_version(0.0)].docx</p> | | |
| Versiones de la Plantilla: | | |
| Versión: | Fecha: | Responsable: |
| 1.0 | 10-noviembre-2014 | Ing. Edgar Montaluisa Ing. Ximena López |

CONTINÚA



| FORMALIZACIÓN DE DESARROLLO DE SOFTWARE | |
|---|--|
| Proyecto: [Nombre del Proyecto] | |
| Fecha de Reunión: [Fecha de Reunión entre las partes] | Documento de Requerimiento: [No. de documento de requerimiento] |
| Unidad Requirente: [Nombre de la Unidad Requirente] | Integrantes Unidad Requirente: [Personal que participará directamente en el proyecto] |
| Unidad Desarrollo: [Nombre de la Unidad Desarrollo] | Integrante Grupo de Desarrollo: [Personal del Grupo de Desarrollo que participará directamente en el proyecto] |
| Descripción: [Relato detallado del requerimiento, que permita establecer el objetivo principal que se persigue con su desarrollo; No incluye requisito solo una descripción global del proyecto a desarrollarse; Debe ser lo suficientemente claro para establecer el alcance global y las iteraciones que serán necesarias para la implementación] | |
| Nombre del Responsable RESPONSABLE UNIDAD REQUIRENTE | Nombre del Líder del Proyecto RESPONSABLE DEL GRUPO DE DESARROLLO |

- Planificación de Entregas.-** Este documento se establecerá como resultado del análisis del requerimiento general, refleja el cronograma de entregas parciales de las versiones resultantes del proceso de

desarrollo, las versiones deberán ajustarse en el tiempo establecido por la metodología, entre 2 y 4 semanas. Es necesario considerar que se ejecutará un proceso completo de desarrollo por cada versión. Este cronograma tendrá la aceptación del grupo de desarrollo y de la unidad requirente, siendo compromiso que a la entrega de cada versión se proceda inmediatamente a trabajar con las funcionalidades desarrolladas. En este documento se detallará de forma resumida que contendrá cada versión a entregar a fin de que sirva de guía para las fases posteriores.

Tabla 4.

Planificación de Entregas

| |
|--|
| <p>Plantilla:</p> <p>“PLANIFICACIÓN DE ENTREGAS”</p> |
| <p>Función de la Plantilla:</p> <p>Esta plantilla establecerá los tiempos en los cuales se realizará la entrega de las versiones de software planificadas en el desarrollo.</p> |
| <p>Objetivos de la Plantilla:</p> <ul style="list-style-type: none"> • Especificar los tiempos estimados en los cuales se entregarán las versiones de software. • Preparar a la Unidad Requirente para el proceso de inducción y organizar su trabajo para dar tiempo a la capacitación del nuevo sistema. • Organizar el trabajo del personal técnico para el cumplimiento de los tiempos estimados en el desarrollo de software. |
| <p>Alcance de la Plantilla:</p> <p>Esta plantilla se usará luego de haber formalizado el desarrollo de software y previo al proceso iterativo de desarrollo de la aplicación.</p> |
| <p>Notas:</p> <p>Esta plantilla se guardará como un archivo en Formato de Texto MS Word (2007-2010) y su nombre seguirá la nomenclatura siguiente: Planificacion.[nombre_de_proyecto].v[número_version(0.0)].docx</p> |

CONTINÚA



| Versiones de la Plantilla: | | | |
|---|--------------------------|---|--|
| Versión: | Fecha: | Responsable: | |
| 1.0 | 10-noviembre-2014 | Ing. Edgar Montaluisa Ing. Ximena López | |
| PLANIFICACIÓN DE ENTREGAS | | | |
| Proyecto: [Nombre del Proyecto] | | | |
| Fecha Elaboración: [Fecha de Elaboración del Documento] | | Tiempo Total Entrega: [Tiempo Total de Desarrollo en días] | |
| Cronograma de Entrega de Versiones: | | | |
| Versión: | Tiempo Desarrollo | Fecha Entrega: | Resumen de la Entrega: |
| [#Versión] | [En días] | [Fecha] | [Breve descripción de la Funcionalidad a entregar en la versión] |
| Nombre del Responsable RESPONSABLE UNIDAD REQUIRENTE | | Nombre del Líder del Proyecto RESPONSABLE DEL GRUPO DE DESARROLLO | |

1.6.3. Requerimientos

En esta fase, que constituye parte del proceso iterativo de desarrollo, se inicia con la especificación detallada del problema en forma de Historia de Usuario, identificando los requerimientos y las relaciones que permitan llegar a la

comprensión del problema. En esta fase se utilizará el siguiente documento para cada una de las historias de usuarios identificadas:

- **Requisitos Específicos.-** Este documento basado en el formato de Historia de Usuarios sintetiza la información necesaria y suficiente para avanzar en el proceso de desarrollo. En ningún momento se pretenderá que su elaboración tome tanto tiempo que retrase el desarrollo de software, su objetivo será guiar y proporcionar los suministros para el diseño, implementación y pruebas del sistema. Para la especificación del requerimiento se utilizará de un documento que recopile esta necesidad de forma detallada.

Tabla 5.

Requisitos Específicos

| |
|---|
| Plantilla: “REQUISITOS ESPECÍFICOS” |
| Función de la Plantilla: Esta plantilla permitirá realizar el levantamiento de los requisitos funcionales detallando de la mejor manera cada uno de los requerimientos del usuario. |
| Objetivos de la Plantilla: <ul style="list-style-type: none"> • Especificar los requisitos funcionales de cada una de las versiones a desarrollar de la aplicación. • Definir las pruebas a realizarse como comprobación del software realizado, a fin de determinar el grado de implementación del requisito en el sistema. |

CONTINÚA



| | | |
|---|--|--|
| Alcance de la Plantilla: | | |
| Esta plantilla se usará de forma iterativa por cada ciclo que se realice en el desarrollo de una versión específica y será el primero en ser diseñado a fin de proporcionar los parámetros para el desarrollo de la aplicación en las siguientes fases. | | |
| Notas: | | |
| Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) y su nombre seguirá la nomenclatura: Requisitos.[nombre_de_proyecto].v[número_version(0.0)].docx | | |
| Versiones de la Plantilla: | | |
| Versión: | Fecha: | Responsable: |
| 1.0 | 10-noviembre-2014 | Ing. Edgar Montaluisa Ing. Ximena López |
| REQUISITOS ESPECÍFICOS | | |
| Proyecto: | | |
| [Nombre del Proyecto] | | |
| Fecha Elaboración: | Versión de Software: | |
| [Fecha de Elaboración del Documento] | [Número de la Versión de desarrollo de software a la que pertenece el requisito] | |
| ID de Requisito: | Título: | |
| [Número de Requisito en "R"+3 dígitos] | [Nombre que resumen la funcionalidad del requisito] | |

CONTINÚA



| | |
|--|---|
| <p>Descripción: [Relato que detalla el proceso a automatizarse, identificando los actores, documentos de fuente de la información, documentos de salida de información, las validaciones requeridas, los atributos de información que deberán ser considerados. Incluirá un gráfico explicativo de requerirse]</p> | |
| <p>Estimación: [Número de Horas que se estima tomará su automatización]</p> | <p>Usuario: [Usuario quien requiere la funcionalidad del sistema]</p> |
| <p>Prioridad: [Escala en números enteros de 1 a 10 que indica la urgencia de realizar el requerimiento]</p> | <p>Dependencia: [ID de los requisitos de los cuales depende este nuevo requisito para poder ser implementado.]</p> |
| <p>Fecha Revisiones: [Fecha en formato dd-mm-aaaa de las revisiones realizadas, y breve descripción de la motivación de la revisión]</p> | |
| <p>Pruebas: [Se establecerá las pruebas que se realizarán en la aplicación desarrollada para poder verificar la correcta implementación del requisito. Se establecerán los datos de entrada y los resultados esperados]</p> | |
| <p>Usuario Responsable USUARIO DE UNIDAD REQUIRENTE</p> | <p>Nombre del Técnico TÉCNICO RESPONSABLE DEL DESARROLLO</p> |

1.6.4 Diseño

En el diseño se establecerán dos artefactos, los que detallarán aspectos como la interface del sistema y al almacenamiento de información. El modelamiento de las interfaces estándar se realizará en la primera versión, y en las siguientes versiones solamente aquellas que difieran del estándar. El documento de interfaces será único ya que es un repositorio incremental de los diseños que se podrán usar en el desarrollo de la aplicación. El modelamiento de datos será incremental de acuerdo a los requerimientos establecidos en cada versión y a la Planificación de Entregas, en el documento se especificará la instancia correspondiente a cada una de las versiones implementadas. Se establecen dos artefactos para esta fase:

- **Diseño de Interfaces.-** Este documento detallará las interfaces estándar a utilizarse en todas las vistas de mantenimiento de la información, estas serán usadas durante el desarrollo de todas las versiones de software. Además proporcionará la plantilla para la documentación de nuevas interfaces que se vayan presentando en el desarrollo del proyecto y sus versiones. El diseño de la interface guardará concordancia con la utilización de las herramientas de desarrollo a fin de minimizar el paso del diseño a la implantación

Tabla 6.

Diseño de Interfaces

| |
|---|
| Plantilla: “DISEÑO DE INTERFACES” |
| Función de la Plantilla: Esta plantilla detallará la ubicación básica de los componentes en las interfaces a fin de mantener una armonía en el diseño. Los procesos que son recurrentes se modelarán una sola vez, dando la opción de modelar de forma detallada solo las interfaces que requieren mayor complejidad. |
| Objetivos de la Plantilla: |

CONTINÚA



| | | |
|--|-------------------|---|
| <ul style="list-style-type: none"> • Especificar el diseño de las interfaces a utilizar, usando diagramas de bloques indicando el posicionamiento de elementos como texto, cajas de entrada de datos, cajas de selección, imágenes, etc. • Detallar aquellas interfaces que requieran mayor complejidad de interpretación. • Guardar armonía en el diseño de las interfaces | | |
| <p>Alcance de la Plantilla:</p> <p>Esta plantilla se usará de forma iterativa en cada ciclo que se realice en el desarrollo de todas las versiones. Se diseñará luego de identificar si el requisito requiere un nuevo diseño de los anteriores ya existentes. El diseño es general y no específico a cada requerimiento.</p> | | |
| <p>Notas:</p> <p>Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) y su nombre seguirá la nomenclatura: Interfaces. [nombre_de_proyecto]. v[número_version(0.0)].docx</p> | | |
| <p>Versiones de la Plantilla</p> | | |
| Versión: | Fecha: | Responsable: |
| 1.0 | 10-noviembre-2014 | Ing. Edgar Montaluisa Ing. Ximena López |
| | | |
| <p>DISEÑO DE INTERFACES</p> | | |
| <p>Proyecto:</p> <p>[Nombre del Proyecto]</p> | | |
| <p>Fecha Elaboración:</p> <p>[Fecha de Elaboración del Documento]</p> | | <p>Técnico:</p> <p>[Nombre del técnico que propuso el uso de la interface]</p> |
| <p>Título:</p> <p>[Nombre que resumen la funcionalidad de la interface]</p> | | |

CONTINÚA



| |
|---|
| <p>Descripción:</p> <p>[Relato que detalla la concepción de la interface, especificando en qué casos se debe utilizar este diseño. Se acompañará con una imagen que represente en Bloques el diseño]</p> |
| <p>Nombre del Técnico Responsable</p> <p>TÉCNICO RESPONSABLE DEL DESARROLLO</p> |

- **Diseño de Base de Datos.-** El documento establece cada una de las versiones de incremento a la base de datos, capturando de forma gráfica y proporcionando los lineamientos de documentación de las innovaciones implementadas. El documento establecerá la operatividad que proporciona la versión de base de datos en base a los requisitos establecidos, especifica los almacenamientos creados en la versión.

Tabla 7.

Diseño de base de datos

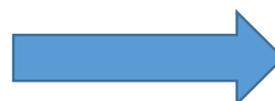
| |
|---|
| <p>Plantilla:</p> <p>“DISEÑO DE BASE DE DATOS”</p> |
| <p>Función de la Plantilla:</p> <p>Detallar la funcionalidad de la base de datos que soportará la versión correspondiente de software desarrollado.</p> |
| <p>Objetivos de la Plantilla:</p> <ul style="list-style-type: none"> • Especificar el modelo de datos que soportará la implementación de una versión completa desarrollada. • Especificar los requerimientos que se han considerado en la versión de la Base de Datos. • Permitir la trazabilidad en la evolución de la Base de Datos |

CONTINÚA



| | | |
|---|--|--|
| Alcance de la Plantilla: | | |
| Esta plantilla se elaborará como resumen del modelamiento de datos, indicando los requisitos implementados y la versión final de la base de datos a ser entregada en la versión del Software. | | |
| Notas: | | |
| Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) y su nombre seguirá la nomenclatura siguiente: Base.[nombre_de_proyecto].v[número_version(0.0)].docx | | |
| Versiones de la Plantilla: | | |
| Versión: | Fecha: | Responsable: |
| 1.0 | 10-noviembre-2014 | Ing. Edgar Montaluisa Ing. Ximena López |
| DISEÑO DE BASE DE DATOS | | |
| Proyecto: | | |
| [Nombre del Proyecto] | | |
| Fecha Elaboración: | Versión del Software: | |
| [Fecha de Elaboración del Documento] | [Número de versión del software a la cual da soporte la Base de Datos] | |
| Requisitos Implementados: | Técnico: | |
| [ID de los Requisitos implementados en la base de datos] | [Nombre del técnico que realizó el modelo de la Base de Datos] | |

CONTINÚA



| |
|--|
| <p>Descripción:</p> <p>[Relato que detalla las funcionalidades a las cuales da soporte la Base de Datos, especificando las implementaciones de la nueva versión. Será necesario anexar la referencia del archivo de modelo de ser necesario, usando el mismo nombre que este documento]</p> |
| <p style="text-align: center;">Nombre del Técnico Responsable</p> <p style="text-align: center;">TÉCNICO RESPONSABLE DEL DESARROLLO</p> |

1.6.5 Implementación

La implementación al ser un proceso de desarrollo de la aplicación requiere la documentación de la versión creada, para lo cual se hará uso del documento:

- **Versión de Software.-** Este documento especificará en número de versión, los requisitos implementados, los parámetros necesarios para la implantación y las ubicaciones de los códigos fuentes. Cada versión incrementará la funcionalidad de la versión anterior y detallará si se han incluido correcciones a versiones anteriores como parte del proceso de desarrollo.

Tabla 8.

Versión de Software

| |
|--|
| <p>Plantilla:</p> <p>“VERSIÓN DE SOFTWARE”</p> |
| <p>Función de la Plantilla:</p> <p>Detallar la funcionalidad que se ha implementado en una versión específica de software, especificando la operatividad y las correcciones realizadas sobre la versión anterior.</p> |

CONTINÚA



| | | |
|--|--|--|
| Objetivos de la Plantilla. | | |
| <ul style="list-style-type: none"> • Detallar las nuevas funcionalidades de la versión desarrollada. • Detallar las correcciones realizadas sobre una versión anterior. • Definir el entorno necesario para la puesta en marcha de la versión de software desarrollada. | | |
| Alcance de la Plantilla: | | |
| Esta plantilla se elaborará como resumen de la implementación de cada versión de software desarrollada, recopilando las funcionalidades y requerimientos implementados. | | |
| Notas: | | |
| Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) y su nombre seguirá la nomenclatura siguiente: Software.[nombre_de_proyecto].v[número_version(0.0)].docx | | |
| Versiones de la Plantilla: | | |
| Versión: | Fecha: | Responsable: |
| 1.0 | 10-noviembre-2014 | Ing. Edgar Montaluisa Ing. Ximena López |
| VERSIÓN DE SOFTWARE | | |
| Proyecto: | | |
| [Nombre del Proyecto] | | |
| Fecha Elaboración: | Versión del Software: | |
| [Fecha de Elaboración del Documento] | [Número de versión del software] | |
| Requisitos Implementados: | Requisitos Corregidos: | |
| [ID de los requisitos que componen esta nueva versión de software] | [ID de los requisitos revisados en esta nueva versión de software] | |

CONTINÚA



| |
|--|
| <p>Descripción:</p> <p>[Relato que detalla las funcionalidades implementadas en la versión de software desarrollado. Descripción de los requerimientos técnicos y operativos para la implantación de la versión. Detallará el repositorio en el cual se encontrará la versión del código fuente de la aplicación]</p> |
| <p style="text-align: center;">Nombre del Técnico Responsable</p> <p style="text-align: center;">TÉCNICO RESPONSABLE DEL DESARROLLO</p> |

1.6.6 Pruebas

La ejecución de la fase de pruebas de función concluirá con la elaboración del documento que se denominará:

- **Pruebas del Sistema.-** Este documento se elaborará luego de haber retroalimentado hacia la fase de desarrollo los fallos detectados y se considere resueltos y superados todos estos. El documento se elaborará al finalizar y haber superado las pruebas, resumiendo las falencias encontradas y los fallos reportados y solucionados. Registrará el tiempo tomado para el proceso de revisión y el tiempo que fue requerido para la solución de fallos.

Tabla 9.

Pruebas de Sistema

| |
|---|
| <p>Plantilla:</p> <p>“PRUEBAS DE SISTEMA”</p> |
| <p>Función de la Plantilla:</p> <p>La función de esta plantilla es detallar los fallos detectados durante la revisión de funcionalidad del software y los resultados de la retroalimentación hacia el grupo de desarrollo, así como las soluciones establecidas.</p> |

CONTINÚA



| | | |
|---|-------------------|--|
| Objetivos de la Plantilla: | | |
| <ul style="list-style-type: none"> • Detallar los fallos detectados sobre la versión final desarrollada. • Retroalimentar las soluciones aplicadas para solventar los fallos detectados. • Establecer el nivel de cumplimiento del proceso de desarrollo de software y la obtención de una versión de software de calidad. | | |
| Alcance de la Plantilla: | | |
| Esta plantilla se elaborará al final de haber realizado las pruebas de funcionalidad del sistema y deberá mostrar un balance positivo sobre los fallos detectados y solucionados. | | |
| Notas: | | |
| Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) y su nombre seguirá la nomenclatura siguiente: Pruebas.[nombre_de_proyecto].v[número_version(0.0)].docx | | |
| Versiones de la Plantilla: | | |
| Versión: | Fecha: | Responsable: |
| 1.0 | 10-noviembre-2014 | Ing. Edgar Montaluisa Ing. Ximena López |
| PRUEBAS DE SISTEMA | | |
| Proyecto: | | |
| [Nombre del Proyecto] | | |
| Fecha Elaboración: | | Versión del Software: |
| [Fecha de Elaboración del Documento] | | [Número de versión del software] |

CONTINÚA



| | | |
|---|-----------------------------------|---|
| Requisitos Validados: [ID de los requisitos que componen esta nueva versión de software y que han sido validados] | | Técnico Revisor: [Nombre y Firma del Técnico encargado de la Revisión del software] |
| Tiempo de Revisión: [Tiempo en número de horas que se utilizaron para la revisión] | | Tiempo de Soluciones: [Tiempo reportado por el Grupo de Desarrollo sobre los fallos reportados] |
| Descripción de Fallos: [Relato que detalla los fallos encontrados y como este afecta al desempeño del sistema, las acciones tomadas para solucionarlas y el estado actual del sistema revisado] | | |
| No. | Fallo: | Solución: |
| [Secuencial de Fallos] | [Descripción del Fallo Reportado] | [Descripción de la Solución Aplicada por el grupo de desarrollo] |
| <p>Nombre del Técnico Responsable</p> <p>TÉCNICO RESPONSABLE DEL DESARROLLO</p> | | |

1.6.7 Puesta en Marcha

La puesta en marcha requerirá la formalización de la entrega de la versión a fin de iniciar con el trabajo por parte del requirente con la versión desarrollada. Al finalizar el proceso de puesta en marcha y capacitación se elaborará el documento:

- **Entrega del Sistema.-** En este documento se detallará la versión, las personas capacitadas, las novedades suscitadas en la capacitación y la implementación del Sistema. Se requiere de la aceptación mediante la firma de los usuarios capacitados y la autoridad requirente. Se compromete al usuario a usar el sistema de forma que permita acoplar

las nuevas versiones de forma continua y se establecen los compromisos a cumplirse en lo referente al registro de información usando el sistema.

Tabla 10.

Entrega del Sistema

| |
|--|
| <p>Plantilla:</p> <p>“ENTREGA DEL SISTEMA”</p> |
| <p>Función de la Plantilla:</p> <p>Detallar los aspectos más importantes del proceso de entrega de la versión, indicando al requirente la operatividad implementada en esta versión del software y contando con la aceptación del mismo.</p> |
| <p>Objetivos de la Plantilla:</p> <ul style="list-style-type: none"> • Detallar las funcionalidades entregadas en la versión del software. • Detallar las novedades de capacitación, tales como: personal capacitado y novedades de la implantación. • Propiciar la continuidad del desarrollo del proyecto, evidenciado la aceptación por parte del responsable del proceso y del técnico responsable del proyecto. |
| <p>Alcance de la Plantilla:</p> <p>Esta plantilla se elaborará al final del ciclo de desarrollo de la versión de software, y en forma posterior al proceso de puesta en marcha y capacitación. De esta manera se finaliza el proceso de desarrollo de la versión y propicia la nueva iteración.</p> |
| <p>Notas:</p> <p>Esta plantilla se guardará como un archivo en Formato de Texto (MS Word 2007-2010) su nombre seguirá la nomenclatura siguiente: Entrega.[nombre_de_proyecto].v[número_version(0.0)].docx</p> |

CONTINÚA



| Versiones de la Plantilla: | | |
|---|-------------------|---|
| Versión: | Fecha: | Responsable: |
| 1.0 | 10-noviembre-2014 | Ing. Edgar Montaluisa Ing. Ximena López |
| ENTREGA DEL SISTEMA | | |
| Proyecto: [Nombre del Proyecto] | | |
| Fecha Elaboración: [Fecha de Elaboración del Documento] | | Versión del Software: [Número de versión del software] |
| Personal Capacitado: [Nombre de las personas y número de horas usadas en la inducción de la nueva versión del software, sumilla del Personal Capacitado] | | Técnico Capacitador: [Nombre del técnico encargado de la implantación y Capacitación] |
| Novedades de Puesta en Marcha y la Capacitación: [Detalle de las novedades suscitadas en la capacitación y el estado de ejecución del sistema por parte de los capacitados] | | |
| Detalle de la Versión: [Funcionalidades implantadas en la versión de Software] | | |
| Nombre del Responsable RESPONSABLE UNIDAD REQUIRENTE | | Nombre del Líder del Proyecto RESPONSABLE DEL GRUPO DE DESARROLLO |

1.6.8 Manuales

El desarrollo de manuales se convierte en un trabajo que no es planificado en el tiempo de desarrollo, razón por la cual se propone que los manuales a ser incluidos en cada proyecto sean los siguientes:

- **Manual de Administrador.-** El manual de administrador deberá contar con las instrucciones necesarias para poner en funcionamiento el sistema, será desarrollado por el técnico encargado de la implantación del sistema, quien debe conocer los requerimientos técnicos de funcionamiento de la plataforma en la que se ejecuta el sistema. Deberá incluir lo siguiente:
 - Proceso de instalación y configuración de la Plataforma Base (Por ejemplo: Apache, PHP, MySQL)
 - Creación de la Base de Datos del Sistema a partir del Script y configuraciones de seguridad.
 - Instalación y configuración de la aplicación.
 - Registro de Usuarios.
 - Inicializaciones importantes para arrancar el sistema.
- **Manual de Usuario.-** El manual de usuario será desarrollado en versiones de acuerdo a las versiones de software que serán liberadas y puestas en marcha. Su desarrollo estará a cargo del técnico encargado de la puesta en marcha aprovechando la retroalimentación por parte del usuario final. En cada versión del manual de usuario deberá contener los siguientes aspectos:
 - Ingreso y salida al sistema
 - Manejo de los menús de opciones
 - Manejo de las interfaces desarrolladas por procesos
 - Manejo de reportes y exportación de datos a otros formatos
 - Índices de búsqueda temática
 - Gráficos ilustrativos

Los manuales estarán disponibles en repositorios en formato PDF que permita un fácil acceso vía internet

CAPÍTULO II

2 DESARROLLO DEL PRODUCTO SOFTWARE

2.1 Formalización de la Necesidad

Mediante reuniones de trabajo y observación directa, se establece el compromiso para el desarrollo de este producto software en la metodología detallada en el capítulo anterior y en cumplimiento de la legislación, procedimiento y artefactos de desarrollo se obtiene el artefacto de formulación de software para el sistema de control de graduados que podemos evidenciar en Anexo 1.

Nota:

Esta planificación se realizó con los requerimientos del Departamento de Eléctrica y Electrónica. Se definen los formatos y procedimientos con los que se va a administrar el seguimiento de graduados.

2.2 Requerimientos

Mediante reuniones de trabajo y observación directa, se establece los requisitos para el desarrollo de este producto software y cumpliendo con la metodología se aplica los artefactos que se encuentran en Anexo 2.

2.3 Diseño

Mediante reuniones de trabajo y observación directa, se establece los requisitos para el desarrollo de este producto software y cumpliendo con la metodología se aplica los artefactos que se encuentran en Anexo 3.

2.4 Implementación

Mediante reuniones de trabajo y observación directa, se establece los requisitos para el desarrollo de este producto software y cumpliendo con la metodología se aplica los artefactos que se encuentran en Anexo 4.

2.5 Pruebas

Mediante reuniones de trabajo y observación directa, se establece los requisitos para el desarrollo de este producto software y cumpliendo con la metodología se aplica los artefactos que se encuentran en Anexo 5.

2.6 Liberación

Mediante reuniones de trabajo y observación directa, se establecen los requisitos para el desarrollo de este producto software y cumpliendo con la metodología se aplica los artefactos que se encuentran en Anexo 6.

CAPÍTULO III

3 MANUAL DE USUARIO Y CAPACITACIÓN AL CLIENTE

El manual del usuario se ha realizado en un video, el mismo que requiere de un reproductor básico de formato MP4, el mismo que se encuentra al servicio de los involucrados del proceso.

La capacitación puntual al cliente queda a la disposición de la Unidad de Tecnología de la Información de ESPE Latacunga, ya que se realiza la entrega formal del sistema, su documentación y su manual de usuario.

CAPÍTULO IV

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Se cumplió el objetivo del proyecto de crear un sistema para la administración de Graduados que nace de la necesidad de contar con datos reales para estadísticas requeridas en el departamento de Electrónica y Eléctrica.
- En el presente tema investigativo el uso del patrón MVC mediante el framework Yii permitió optimizar el tiempo de desarrollo utilizando métodos propios del framework.
- Con la metodología de las TIC's de la Universidad de las Fuerzas Armadas ESPE extensión Latacunga, se mejora el tiempo de diseño y desarrollo porque es una metodología ágil y se puede usar en cualquier proyecto de éstas características.
- Se aplicaron cada una de las fases de desarrollo la cual requiere de la documentación de cada uno de los artefactos que intervinieron en el desarrollo, se llevaron los lineamientos de programación y base de datos garantizando el cumplimiento de los mismos.
- La documentación de cada uno de los artefactos usados en la metodología de las TIC'S, permite que se pueda realizar a futuros mejoras, modificaciones, crear nuevas opciones según crezcan las necesidades de los usuarios del sistema.

4.2 Recomendaciones

- Se recomienda usar software libre para desarrollos de este tipo de proyectos, ya que cualquier persona puede usar bajo condiciones de la licencia, es de bajo costo y existe libertad de conocimiento y trabajo cooperativo a través de blogs interactivos en la web.
- Se sugiere aplicar la metodología de las TIC's de la Universidad de las Fuerzas Armadas ESPE extensión Latacunga por ser una metodología ágil y se puede usar en cualquier proyecto de estas características.
- Se recomienda tener un repositorio único de código fuente para evitar que se sobrescriban archivos.

- Se recomienda tener un servidor de compilación continua, de ésta forma garantizar las publicaciones de versiones con todas las clases, servicios y dll actualizadas.
- Se recomienda el uso de base de datos libre como MySQL o Postgresql ya que es posible desarrollar propios tipos de datos, permite la innovación y libertad en la programación, velocidad al realizar las operaciones lo que hace uno de los gestores con mejor rendimiento, no se requiere de un equipo de costosas características. Su conectividad, velocidad y seguridad hacen que MySQL Server sea muy apropiado para acceder bases de datos en Internet.
- Se recomienda contratar un certificado de seguridad SSL (Secure Socket Layer) con la finalidad de complementar la seguridad de la aplicación en la transferencia de datos entre el navegador y el servidor Web.
- Se recomienda actualizar la malla de formación de tal forma que los estudiantes perciban y vivan una experiencia real del desarrollo de software en ingeniería y con buenas prácticas.

REFERENCIAS BIBLIOGRÁFICAS

- Caballero, L., & Lara, J. (2006). Tercer Congreso Nacional Software Libre Venezuela. Merida.
- Erich, G., Helm, R., Johnson, R., & Vlissides, J. (2003). *Patrones de Diseño*. Madrid: Addison Wesley.
- Gomaa, H. (2011). *Software modeling and design uml use cases patterns and software architectures*. Cambridge.
- Gutiérrez., J. J. (07 de 05 de 2006). *Lenguajes y Sistemas Informático*. Recuperado el 21 de 04 de 2015, de Lenguajes y Sistemas Informático: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- Ivar, J. (2004). *Aspect-Oriented Software Development With Use Cases*. Massachusetts: Addison-Wesley Professional .
- Kruchten, P. (1995). Modelo de arquitectura. *IEEE Software*, 45-50.
- López, X., & Montaluisa, F. (2015). *Diseño de la estructura organizativa e implementación de una metodología para el desarrollo de software en la Fábrica de Software de la ESPE Extensión Latacunga*. Latacunga: Universidad de las Fuerzas Armadas - ESPE.
- Mann, M. (1997). *Ingeniería de Software*. massachusetts: Mikel Angoar.

ANEXOS

Anexo 1. Formalización de la necesidad

| FORMALIZACIÓN DE DESARROLLO DE SOFTWARE | |
|--|--|
| Proyecto: Sistema de Control de Seguimiento de Graduados y Graduadas | |
| Fecha de Reunión: 2014-02-19 | Documento de Requerimiento: Memorándum 2014-0013-ESPE-SL-e-09 2014-02-10 de la Unidad de Admisión y Registro |
| Unidad Requirente: Universidad de las Fuerzas Armadas ESPE – Extensión Latacunga | Integrantes Unidad Requirente: Ing. Nancy Guerrón |
| Unidad Desarrollo: Grupo de Desarrollo de Tesis | Integrante Grupo de Desarrollo: Adela Sofía Gallardo Cueva |
| Descripción: El proyecto tiene como finalidad el conocer mediante técnicas apropiadas el proceso de seguimiento de graduados y graduadas de esta forma generar un producto software (análisis, diseño e implementación) que permitirá al administrador del proceso hacer un seguimiento minucioso efectivo y oportuno logrando así agilizar el proceso que actualmente es muy escaso ya que no existe un sistema que contemple los nuevos requerimientos de la LOES. | |
| Ing. Nancy Guerrón  RESPONSABLE UNIDAD REQUIRENTE | Sofía Gallardo  RESPONSABLE DEL GRUPO DE DESARROLLO |

PLANIFICACIÓN DE ENTREGAS

Proyecto:

Sistema de Control de Seguimiento de Graduados y Graduas

Fecha Elaboración:

2014-05-19

Tiempo Total Entrega:

60 días

Cronograma de Entrega de Versiones:

| Versión: | Tiempo Desarrollo | Fecha Entrega: | Resumen de la Entrega: |
|-----------------|--------------------------|-----------------------|---|
| 1.0 | 10 días | 2014-05-30 | Definición de menús, diseño de las tablas Departamento, Carrera, Coordinador de Egresado y Empresa. |
| 2.0 | 15 días | 2014-06-20 | Diseño de las tablas Graduado, Dirección, Experiencia Laboral, Preguntas, Título Adicional. |
| 3.0 | 20 días | 2014-07-18 | Desarrollo del proceso para la Encuesta. |
| 4.0 | 15 días | 2014-08-01 | Definición y desarrollo de Reportes requeridos. |

Ing. Nancy Guerrón



RESPONSABLE UNIDAD
REQUIRENTE

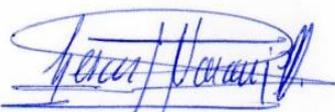
Sofía Gallardo



RESPONSABLE DEL GRUPO DE
DESARROLLO

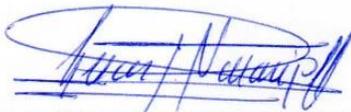
Anexo 2. Requerimientos

| REQUISITOS ESPECIFICOS | |
|--|---|
| Proyecto: Sistema de Control de Seguimiento de Graduados y Graduas | |
| Fecha Elaboración: 2014-05-19 | Versión de Software: 1.0 |
| ID de Requisito: R001-V1.0 | Título: Diseño de menús |
| Descripción: Se debe realizar el diseño de los menús que va a utilizar el sistema, siendo de fácil localización todos los procesos dentro del sistema y según el perfil de usuario registrado. | |
| Estimación: 8 horas | Usuario: Ing. César Naranjo |
| Prioridad: 1 | Dependencia: N/A |
| Fecha Revisiones: 2014-05-22 Se realiza la revisión de la estructura de los menús. 2014-05-30 Se realiza la revisión de la estructura de los menús debido a la actualización de la misma. | |
| Pruebas: El menú deberá ser de fácil uso para el usuario, pudiendo identificar de manera clara todos los módulos de la aplicación. | |
| Ing. César Naranjo  USUARIO DE UNIDAD REQUIRENTE | Sofía Gallardo  TÉCNICO RESPONSABLE DEL DESARROLLO |

| REQUISITOS ESPECIFICOS | |
|---|---|
| Proyecto: Sistema de Control de Seguimiento de Graduados y Graduas | |
| Fecha Elaboración: 2014-05-19 | Versión de Software: 1.0 |
| ID de Requisito: R002-V1.0 | Título: Diseño de la tabla Departamento |
| Descripción: <p>Se debe realizar el ingreso de los departamentos existentes, contando con ID que deberá ser numérico y auto incremental, ID de banner deberá ser alfanumérico, nombre del departamento deberá ser alfanumérico y nombre del director de este departamento que de igual forma deberá ser alfanumérico. Todos estos datos, exceptuando el ID de banner, son obligatorios.</p> | |
| Estimación: 10 horas | Usuario: Ing. César Naranjo |
| Prioridad: 2 | Dependencia: R001.V1.0 |
| Fecha Revisiones: <p>2014-05-23 Se realiza la revisión de la operatividad de los departamentos.</p> | |
| Pruebas: <p>Se realizará las pruebas registrando los departamentos en el sistema con cada uno de los datos existentes.</p> | |
| Ing. César Naranjo  USUARIO DE UNIDAD REQUIRENTE | Sofía Gallardo  TÉCNICO RESPONSABLE DEL DESARROLLO |

| REQUISITOS ESPECIFICOS | |
|--|--|
| Proyecto: Sistema de Control de Seguimiento de Graduados y Graduas | |
| Fecha Elaboración: 2014-05-19 | Versión de Software: 1.0 |
| ID de Requisito: R003-V1.0 | Título: Diseño de la tabla de Carreras |
| <p>Descripción:</p> <p>Se debe realizar el ingreso de las carrera existentes, contando con ID que deberá ser numérico y auto incremental, ID de Banner deberá ser alfanumérico, tipo de carrera que podrá ser tecnología o ingeniería; por ende; deberá ser alfanumérico, nombre de la carrera deberá ser alfanumérico, nombre del director de la carrera que deberá ser alfanumérico, número de cedula del director de la carrera que deberá ser alfanumérico, siglas de la carrera que será alfanumérico, mínimo de créditos para poder cumplir la malla curricular del graduado será numérico y el departamento al que pertenece este campo deberá ser numérico y será la clave foránea para relacionar con los departamentos. Todos los campos, exceptuando ID de banner, son obligatorios.</p> | |
| Estimación: 26 horas | Usuario: Ing. César Naranjo |
| Prioridad: 3 | Dependencia: R001-V1.0 |
| <p>Fecha Revisiones:</p> <p>2014-05-28 Se realiza la revisión del funcionamiento de las carreras.</p> | |
| <p>Pruebas:</p> <p>Se realizará las pruebas registrando las carreras en el sistema con cada uno de los datos existentes.</p> | |
| <p>Ing. César Naranjo</p>  <p>USUARIO DE UNIDAD REQUIRENTE</p> | <p>Sofía Gallardo</p>  <p>TÉCNICO RESPONSABLE DEL DESARROLLO</p> |

| REQUISITOS ESPECIFICOS | |
|---|--|
| Proyecto: Sistema de Control de Seguimiento de Graduados y Graduas | |
| Fecha Elaboración: 2014-05-19 | Versión de Software: 1.0 |
| ID de Requisito: R004-V1.0 | Título: Diseño de la tabla Coordinador de Egresado |
| Descripción: Se debe realizar el ingreso de los coordinadores de graduados contando con ID que será numérico y auto incremental, ID de banner que deberá ser alfanumérico, cédula que será alfanumérico y validando que sea una cédula correcta, nombre del coordinador que será alfanumérico, apellido del coordinador que será alfanumérico, correo del coordinador validando que el mismo sea correcto este será alfanumérico, teléfono del coordinador que será alfanumérico y carrera que será numérico y campo foráneo que relacionara con cada una de las carreras existentes. | |
| Estimación: 10 horas | Usuario: Ing. César Naranjo |
| Prioridad: 4 | Dependencia: R001.V1.0 R002.V1.0 R003.V1.0 |
| Fecha Revisiones: 2014-05-28 Se realiza la revisión del funcionamiento de la administración del coordinador de graduados. | |
| Pruebas: Se realizará las pruebas registrando los coordinadores por carrera y departamento en el sistema con cada uno de los datos existentes. | |
| Ing. César Naranjo  USUARIO DE UNIDAD REQUERENTE | Sofía Gallardo  TECNICO RESPONSABLE DEL DESARROLLO |

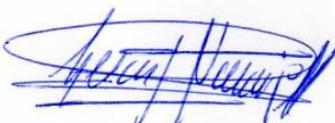
| REQUISITOS ESPECIFICOS | |
|--|---|
| Proyecto: Sistema de Control de Seguimiento de Graduados y Graduas | |
| Fecha Elaboración: 2014-05-19 | Versión de Software: 1.0 |
| ID de Requisito: R005-V1.0 | Título: Diseño de la tabla Empresa. |
| Descripción: <p>Se debe realizar el ingreso de las empresas, contando con ID que deberá ser numérico y auto incremental, ID de banner que deberá ser alfanumérico, nombre de la empresa que deberá ser alfanumérico, actividad de la empresa que deberá ser alfanumérico, dirección de la empresa que deberá ser alfanumérico, representante legal de la empresa que deberá ser alfanumérico, cargo del representante legal de la empresa que deberá ser alfanumérico, título del representante legal de la empresa que deberá ser alfanumérico, teléfono de la empresa que deberá ser alfanumérico y correo electrónico de la empresa que de igual forma deberá ser alfanumérico. Todos los campos, exceptuando ID de banner, son obligatorios.</p> | |
| Estimación: 20 horas | Usuario: Coordinador de Graduados |
| Prioridad: 5 | Dependencia: R001.V1.0 |
| Fecha Revisiones: <p>2014-05-30 Se realiza la revisión del funcionamiento de las empresas.</p> | |
| Pruebas: <p>Se realizarán el ingreso de información de las empresas registrando cada una e identificando los campos obligatorios de cada una tanto en ingreso como en modificaciones.</p> | |
| Ing. César Naranjo  USUARIO DE UNIDAD REQUIRENTE | Sofía Gallardo  TÉCNICO RESPONSABLE DEL DESARROLLO |

| REQUISITOS ESPECIFICOS | |
|---|--|
| Proyecto: Sistema de Control de Seguimiento de Graduados y Graduas | |
| Fecha Elaboración: 2014-05-19 | Versión de Software: 2.0 |
| ID de Requisito: R001-V2.0 | Título: Diseño de la tabla Graduado |
| Descripción: Se debe realizar el ingreso del graduado contando con ID que será numérico y auto incremental, ID de banner que será alfanumérico, cédula que será alfanumérico y validando que sea una cédula correcta, nombre del graduado que será alfanumérico, apellido del graduado que será alfanumérico, correo del graduado validando que el mismo sea correcto este será alfanumérico, teléfono del graduado que será alfanumérico y carrera a la que pertenece el graduado que deberá ser numérico y campo foráneo para relacionarlo con las carreras. Todos los campos son obligatorios. | |
| Estimación: 14 horas | Usuario: Ing. César Naranjo |
| Prioridad: 6 | Dependencia: R001-V1.0 R002-V1.0 R003-V1.0 R004-V1.0 R005-V1.0 |
| Fecha Revisiones: 2014-06-23 Se realiza la revisión del funcionamiento de los graduados. | |
| Pruebas: Se realizará las pruebas registrando cada graduado y relacionándolo con su carrera correspondiente. | |
| Ing. César Naranjo  USUARIO DE UNIDAD REQUIRENTE | Sofía Gallardo  TÉCNICO RESPONSABLE DEL DESARROLLO |

| REQUISITOS ESPECIFICOS | |
|---|---|
| Proyecto: Sistema de Control de Seguimiento de Graduados y Graduadas | |
| Fecha Elaboración: 2014-05-19 | Versión de Software: 2.0 |
| ID de Requisito: R002-V2.0 | Título: Diseño de las tablas Domicilio |
| Descripción: <p>Se debe realizar el ingreso de la dirección del domicilio del graduado, si está trabajando debe ingresar la dirección de su trabajo actual. Se registra la ciudad, provincia, calle principal, número de domicilio y calle secundaria, referencia. Todos los campos obligatorios, los mismos datos para el domicilio del trabajo.</p> | |
| Estimación: 10 horas | Usuario: Ing. César Naranjo |
| Prioridad: 7 | Dependencia: R006-V1.0 |
| Fecha Revisiones: <p>2014-06-23 Se realiza la revisión del funcionamiento del ingreso del domicilio del graduado</p> | |
| Pruebas: <p>Se realizara las pruebas ingresando la dirección del domicilio y si trabaja debe ingresar la dirección de su trabajo.</p> | |
| Ing. César Naranjo  USUARIO DE UNIDAD REQUIRENTE | Sofía Gallardo  TÉCNICO RESPONSABLE DEL DESARROLLO |

| | |
|---|--|
| REQUISITOS ESPECIFICOS | |
| Proyecto: Sistema de Control de Seguimiento de Graduados y Graduas | |
| Fecha Elaboración: 2014-05-19 | Versión de Software: 2.0 |
| ID de Requisito: R003-V2.0 | Título: Diseño de las tablas Experiencia Laboral. |
| Descripción: Se va a diseñar los campos requeridos para registrar la experiencia laboral, la historia laboral queda registrada por período y asociada al id del graduado. | |
| Estimación: 10 horas | Usuario: Ing. César Naranjo |
| Prioridad: 8 | Dependencia: R001-V1.0 |
| Fecha Revisiones: 2014-06-23 Se realiza la revisión del funcionamiento de los graduados. | |
| Pruebas: Se realizará las pruebas registrando cada graduado ingresando si tiene experiencia laboral | |
| <p>Ing. César Naranjo</p>  <p>USUARIO DE UNIDAD REQUIRENTE</p> | <p>Sofía Gallardo</p>  <p>TÉCNICO RESPONSABLE DEL DESARROLLO</p> |

| REQUISITOS ESPECIFICOS | |
|--|--|
| Proyecto: Sistema de Control de Seguimiento de Graduados y Graduas | |
| Fecha Elaboración: 2014-05-19 | Versión de Software: 2.0 |
| ID de Requisito: R004-V2.0 | Título: Diseño de la tabla Pregunta |
| Descripción: <p>Se van a diseñar la tabla que permitirá alimentar información adicional del graduado, se crea con id asociado al id del graduado y poder completar la información de la dirección y experiencia laboral del graduado, título adicional, empresa donde labora y el historial laboral.</p> | |
| Estimación: 10 horas | Usuario: Ing. César Naranjo |
| Prioridad: 9 | Dependencia: N/A |
| Fecha Revisiones: <p>2014-06-23 Se realiza la revisión de las preguntas que contiene la encuesta para ingresar la información al sistema</p> | |
| Pruebas: <p>Se realizará las pruebas ingresando un graduado con toda la información requerida en el sistema.</p> | |
| <p>Ing. César Naranjo</p>  <p>USUARIO DE UNIDAD REQUIRENTE</p> | <p>Sofía Gallardo</p>  <p>TÉCNICO RESPONSABLE DEL DESARROLLO</p> |

| REQUISITOS ESPECIFICOS | |
|--|---|
| Proyecto: Sistema de Control de Seguimiento de Graduados y Graduas | |
| Fecha Elaboración: 2014-05-19 | Versión de Software: 3.0 |
| ID de Requisito: R001-V3.0 | Título: Desarrollo del proceso para la Encuesta. |
| Descripción: Se va a diseñar la encuesta que permitirá ingresar las preguntas que alimentará los datos de empresa, dirección laboral, títulos adicionales, experiencia laboral. | |
| Estimación: 10 horas | Usuario: Ing. César Naranjo |
| Prioridad: 1 | Dependencia: R001-V2.0 R002-V2.0 R003-V2.0 R004-V2.0 R005-V2.0 R006-V2.0 |
| Fecha Revisiones: 2014-06-23 Se realiza la revisión de las preguntas de la encuesta que serán cargadas al sistema, y sus dependencias para identificar que tablas serán alimentadas en el proceso. | |
| Pruebas: Se cargarán datos de graduados, con todo el proceso de encuesta. | |
| Ing. César Naranjo  USUARIO DE UNIDAD REQUIRENTE | Sofía Gallardo  TÉCNICO RESPONSABLE DEL DESARROLLO |

REQUISITOS ESPECIFICOS

Proyecto: Sistema de Control de Seguimiento de Graduados y Graduadas

Fecha Elaboración:

2014-05-19

Versión de Software:

4.0

ID de Requisito:

R001-V4.0

Título:

Diseño de los formatos para impresión.

Descripción:

Se debe realizar los reportes según lo correspondiente a:

Reporte – Resumen de Graduados por Carrera

| ORD. | CARRERA | REGISTRADOS |
|-------|---|-------------|
| 1 | Ingeniería en Software | 5 |
| 2 | Ingeniería Electrónica e Instrumentación | 2 |
| 3 | Ingeniería Electromecánica | 0 |
| 4 | Tecnología en Computación | 0 |
| 5 | Tecnología en Electromecánica | 0 |
| 6 | Tecnología en Electrónica | 0 |
| 7 | Ingeniería Automotriz | 0 |
| 8 | Ingeniería en Mecatrónica | 0 |
| 9 | Ingeniería en Petroquímica | 1 |
| 10 | Tecnología Automotriz | 0 |
| 11 | Ingeniería Comercial | 1 |
| 12 | Ingeniería en Finanzas y Auditoría | 1 |
| 13 | Ingeniería en Administración Turística y Hotelera | 1 |
| TOTAL | | 11 |

Reporte – Graduados por Carrera

| ORD. | NOMBRE | CEDULA | TELEFONO | CORREO |
|------|-------------------------------|------------|-----------|------------------------|
| 1 | ADELA SOFIA GALLARDO CUEVA | 0501803134 | FEMENINO | asofia@c@hotmail.com |
| 2 | JOHNNY LEON | 0501633640 | FEMENINO | johnny@correo.com.ec |
| 3 | CARLOS MARCELO SILVA MONTEROS | 0500291554 | MASCULINO | marcelo@correo.com |
| 4 | VERÓNICA DE LA CRUZ | 0603224982 | FEMENINO | verito@hotmail.com |
| 5 | ADELA SOFIA ALVAREZ LEON | 0501803134 | MASCULINO | asofia@c@hotmail.com |
| 6 | JUAN MANUEL SORIA MOLINA | 1803282266 | MASCULINO | Juan_Soria@hotmail.com |

Reporte – Ficha de Graduado

ESPE SISTEMA DE GESTIÓN DE EGRESADOS Inicio | Salir (admin)

Graduado | **Reporte** | Estadísticas | Administración Usuarios | Ayuda de

Carrera: **ESPE HARAUJO**

| DATOS PERSONALES | |
|------------------|-------------------|
| CEDELA: | 0501490505 |
| APELLIDOS: | HARAUJO |
| NOMBRES: | CESAR |
| GENERO: | MASCULINO |
| EMAIL: | cesar@espe.edu.ec |

| DIFERENCIACIÓN ACADÉMICA | |
|--------------------------|--|
| CARRERA: | Ingeniería Electrónica e Instrumentación |

| EXPERIENCIA LABORAL | |
|---------------------|----------------|
| EMPRESA: | ESPE LATACUNGA |
| PUESTO: | COORDINADOR |
| TIEMPO: | 24 AÑOS |
| SUELDO: | 3000.00 |

Copyright © 2013
DESARROLLADO POR TOMA CALLARDO

Reporte – Detalle de Asignatura Representativa

ESPE SISTEMA DE GESTIÓN DE EGRESADOS Inicio | Salir (admin)

Graduado | **Reporte** | Estadísticas | Administración Usuarios | Ayuda de

Carrera: **Ingeniería Electrónica e Instrumentación**

Detalle de Asignatura Representativa

REPORTE DETALLADO DE ASIGNATURA REPRESENTATIVA

| ORD. | ASIGNATURA |
|------|---|
| 1 | BASE DE DATOS, PROGRAMACION |
| 2 | ATENCIÓN AL CLIENTE |
| 3 | REDES DE COMUNICACIONES, SISTEMAS DE COMUNICACIONES |
| 4 | TELEFONIA Y REDES DE DATOS |

Copyright © 2013
DESARROLLADO POR TOMA CALLARDO

Reporte – Detalle de Asignatura a Analizar

ESPE SISTEMA DE GESTIÓN DE EGRESADOS Inicio | Salir (admin)

Graduado | **Reporte** | Estadísticas | Administración Usuarios | Ayuda de

Carrera: **Ingeniería Electrónica e Instrumentación**

Detalle de Asignatura Analizar

REPORTE DETALLADO DE ASIGNATURA ANALIZAR

| ORD. | ASIGNATURA |
|------|---------------------|
| 1 | PROGRAMACION WEB |
| 2 | PROGRAMACION |
| 3 | REDES INALAMBRICAS |
| 4 | PREDICCIÓN DE CAMPO |

Copyright © 2013
DESARROLLADO POR TOMA CALLARDO

Reporte – Detalle de Sueldos por Empresa y Puesto

ESPE SISTEMA DE GESTIÓN DE EGRESADOS

Inicio | Salir Admin

Operaciones

Detalle de Sueldos por Empresa y Puesto

REPORTES DETALLADO DE SUELDOS

CARRERA: 120112033

| OBD | EMPRESA | POSTO DE PUESTO | SUELDO |
|-----|-------------|-----------------|--------|
| 1 | UNIVERSIDAD | PROFESOR | 1200 |
| 2 | UNIVERSIDAD | PROFESOR | 1200 |

Copyright © 2015
DEPARTAMENTO DE TICS Y TICX CALLAO

Estadística - Graduados por Carrera

ESPE SISTEMA DE GESTIÓN DE EGRESADOS

Inicio | Salir Admin

Operaciones

Resumen de Graduados por Carrera

REPORTES GENERAL DE GRADUADOS POR CARRERA

| OBD | CARRERA | REGISTRADOS |
|-----|---|-------------|
| 1 | Ingeniería en Software | 15 |
| 2 | Ingeniería Electrónica e Instrumentación | 10 |
| 3 | Ingeniería Electromecánica | 8 |
| 4 | Electrónica en Comunicaciones | 5 |
| 5 | Electrónica en Electromecánica | 5 |
| 6 | Electrónica en Mecatrónica | 5 |
| 7 | Ingeniería Asistencial | 5 |
| 8 | Ingeniería en Mecatrónica | 5 |
| 9 | Ingeniería en Robótica | 5 |
| 10 | Ingeniería en Mecatrónica | 5 |
| 11 | Ingeniería en Física y Aplicada | 5 |
| 12 | Ingeniería en Física y Aplicada | 5 |
| 13 | Ingeniería en Administración, Turismo y Hotelería | 5 |
| 14 | TOTAL | 80 |

Copyright © 2015
DEPARTAMENTO DE TICS Y TICX CALLAO

Estadística – Resumen Información Personal

ESPE SISTEMA DE GESTIÓN DE EGRESADOS

Inicio | Salir Admin

Operaciones

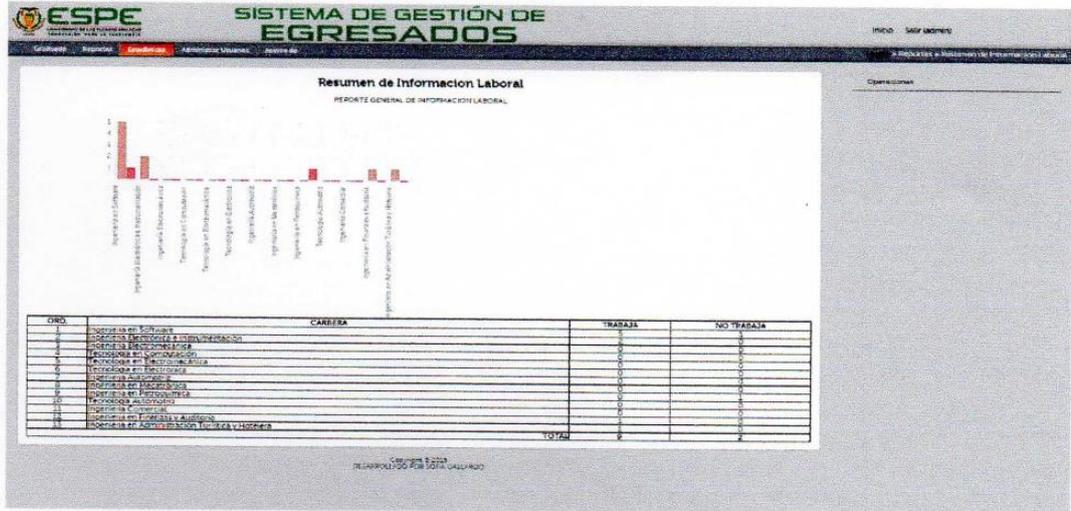
Resumen de Información Personal

REPORTES GENERAL DE INFORMACIÓN PERSONAL

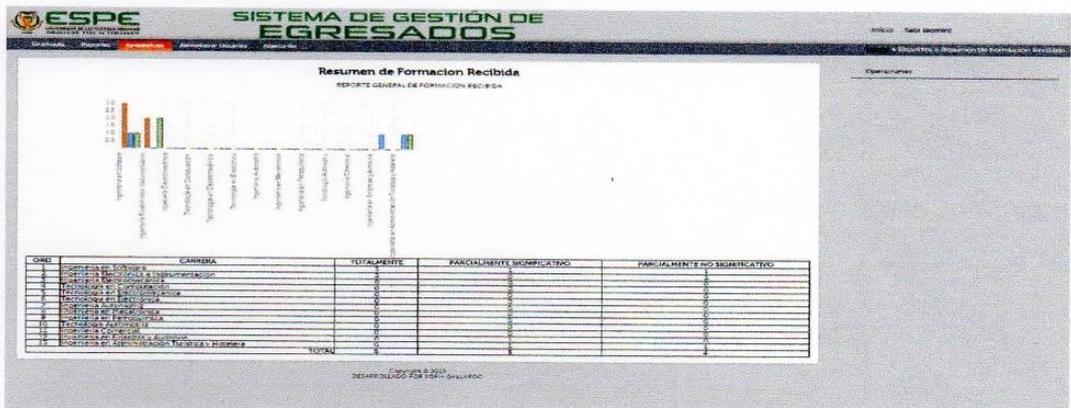
| OBD | CARRERA | HOMBRE | MUJER |
|-----|---|--------|-------|
| 1 | Ingeniería en Software | 10 | 5 |
| 2 | Ingeniería Electrónica e Instrumentación | 8 | 2 |
| 3 | Ingeniería Electromecánica | 6 | 2 |
| 4 | Electrónica en Comunicaciones | 4 | 1 |
| 5 | Electrónica en Electromecánica | 4 | 1 |
| 6 | Electrónica en Mecatrónica | 4 | 1 |
| 7 | Ingeniería Asistencial | 4 | 1 |
| 8 | Ingeniería en Mecatrónica | 4 | 1 |
| 9 | Ingeniería en Robótica | 4 | 1 |
| 10 | Ingeniería en Mecatrónica | 4 | 1 |
| 11 | Ingeniería en Física y Aplicada | 4 | 1 |
| 12 | Ingeniería en Física y Aplicada | 4 | 1 |
| 13 | Ingeniería en Administración, Turismo y Hotelería | 4 | 1 |
| 14 | TOTAL | 70 | 10 |

Copyright © 2015
DEPARTAMENTO DE TICS Y TICX CALLAO

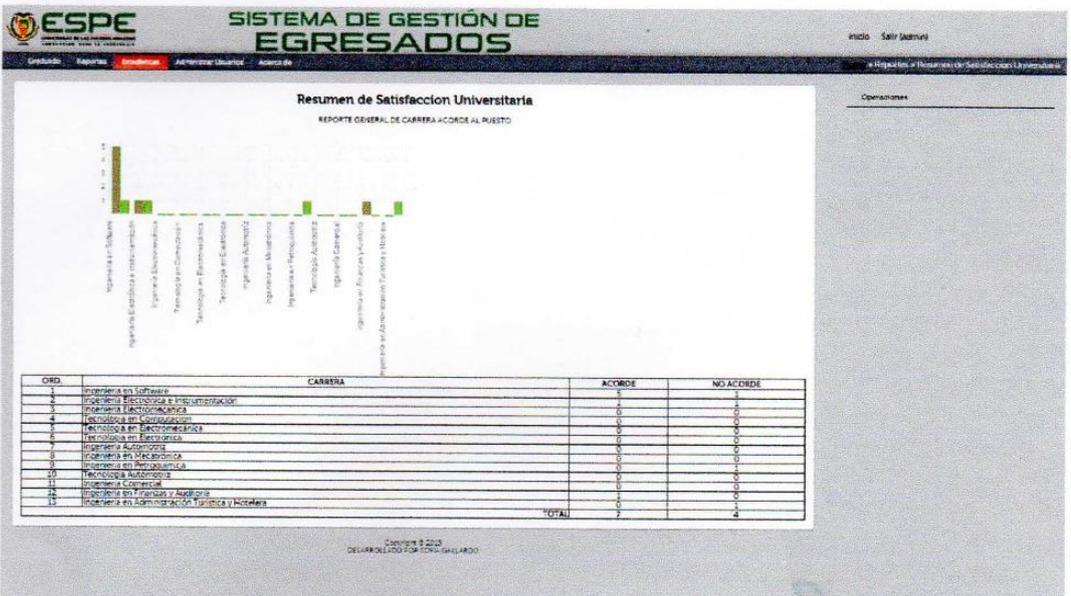
Estadística – Resumen de Información Laboral

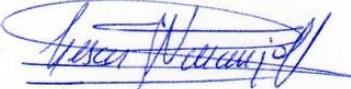


Estadística – Resumen de Formación Recibida



Estadística – Formación Universitaria



| | |
|--|--|
| Estimación: 48 horas | Usuario: Ing. César Naranjo |
| Prioridad: 1 | Dependencia: R001-V1.0 R002-V1.0 R003-V1.0 R004-V1.0 R005-V1.0 R005-V1.0 R001-V2.0 R002-V2.0 R003-V2.0 R004-V2.0 R005-V2.0 R001-V3.0 |
| Fecha Revisiones: | |
| 2014-08-08 Se realiza la revisión del funcionamiento de cada uno de los formatos. | |
| Pruebas: | |
| Se probarán las impresiones de dichos formatos revisando si los datos contenidos en los mismos corresponden a la información ingresada de los graduados. | |
| Ing. César Naranjo  USUARIO DE UNIDAD REQUIRENTE | Sofía Gallardo  TÉCNICO RESPONSABLE DEL DESARROLLO |

Anexo 3. Diseño

| | |
|--|--------------------------------|
| DISEÑO DE INTERFACES | |
| Proyecto: Sistema de Control de Seguimiento de Graduados y Graduas | |
| Fecha Elaboración: 2014-05-19 | Técnico: Sofía Gallardo |
| Título: Ingreso de los datos del Graduado, usando las preguntas de encuesta y completando la información de empresas, experiencia laboral y título adicional. | |
| Descripción: Seleccionamos del menú principal la opción ingreso y ésta despliega la ventana para ingresar los datos personales. Estos datos son: nombres completos, apellidos, etnia, género, estado civil, número de dependientes, cédula, correo, fecha de nacimiento. Luego se carga la ventana para ingresar la dirección personal con los siguientes datos: calle principal, secundaria, el número el sector, y alguna referencia. A continuación se despliegan las preguntas cargadas en la encuesta. En la pregunta referente a la opción si trabaja se despliega la pantalla para ingresar los datos de la empresa: Nombre, denominación social, ruc, giro del negocio de la empresa, representante legal, dirección, teléfono, referencia. Continúa las preguntas y aparece si ha tenido trabajos anteriores, con lo cual se comienza a levantar información histórica laboral del graduado. Al seguir ingresando los datos de la encuesta en la sección si continuó sus estudios se despliega la ventana para ingresar los datos de la institución educativa: nombre, dirección, área de estudio, duración, datos del grado, año de la promoción, título o grado obtenido. Al grabar los datos se genera el id del graduado ingresado. Para modificación de la información se selecciona la opción en el menú y se procede a navegar en las pantallas y hacer la modificación correspondiente. De esta información se generarán las interfaces para los reportes establecidos en el requerimiento 4. | |
| Sofía Gallardo  | |
| TÉCNICO RESPONSABLE DEL DESARROLLO | |

DISEÑO DE BASE DE DATOS

Proyecto:

Sistema de Control de Seguimiento de Graduados y Graduas

Fecha Elaboración:

2014-05-19

Versión del Software:

1.0

Requisitos Implementados:

R001-V1.0

R002-V1.0

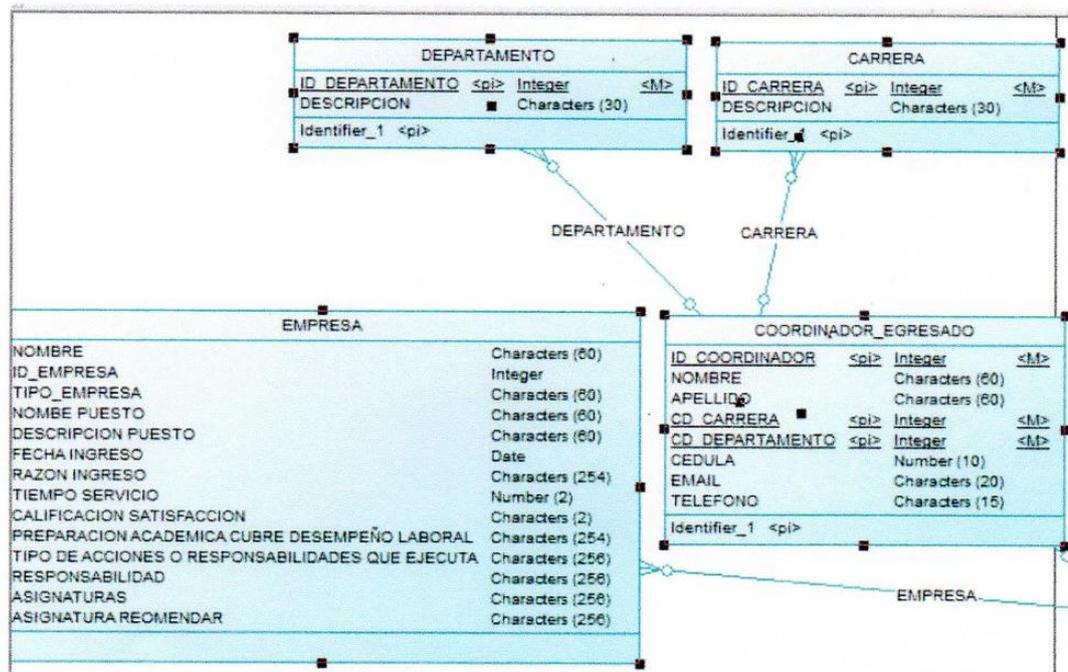
R003-V1.0

R004-V1.0

Técnico:

Sofía Gallardo

Descripción:



Sofía Gallardo

TÉCNICO RESPONSABLE DEL DESARROLLO

DISEÑO DE BASE DE DATOS

Proyecto:

Sistema de Control de Seguimiento de Graduados y Graduas

Fecha Elaboración:

2014-05-19

Versión del Software:

2.0

Requisitos Implementados:

R001-V2.0 R002-V2.0

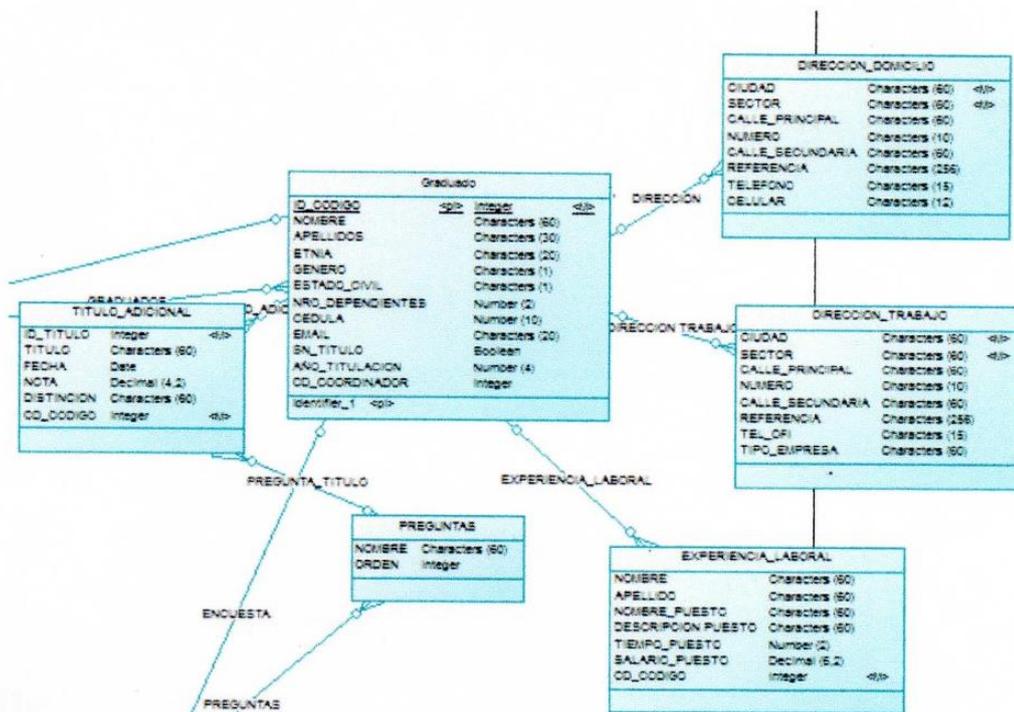
R003-V2.0 R004-V2.0

R005-V2.0

Técnico:

Sofía Gallardo

Descripción:



Sofía Gallardo

TÉCNICO RESPONSABLE DEL DESARROLLO

DISEÑO DE BASE DE DATOS

Proyecto:

Sistema de Control de Seguimiento de Graduados y Graduas

Fecha Elaboración:

2014-05-19

Versión del Software:

3.0

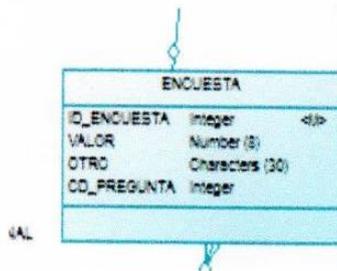
Requisitos Implementados:

R001-V3.0

Técnico:

Sofía Gallardo

Descripción:



Sofía Gallardo

TÉCNICO RESPONSABLE DEL DESARROLLO

DISEÑO DE BASE DE DATOS

Proyecto:

Sistema de Control de Seguimiento de Graduados y Graduas

Fecha Elaboración:

2014-05-19

Versión del Software:

4.0

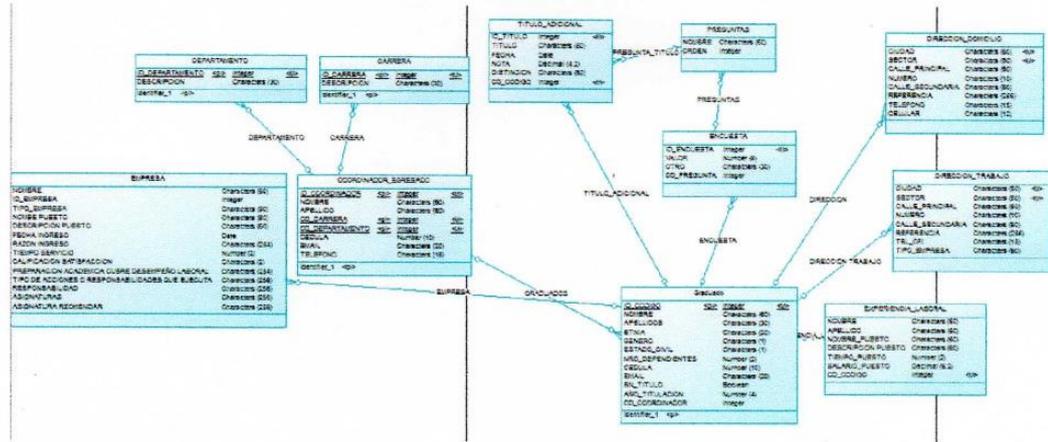
Requisitos Implementados:

R001-V4.0

Técnico:

Sofía Gallardo

Descripción:



Sofía Gallardo

TÉCNICO RESPONSABLE DEL DESARROLLO

Anexo 4. Implementación

| VERSION DE SOFTWARE | |
|---|--------------------------------------|
| Proyecto: Sistema de Control de Graduados y Graduadas | |
| Fecha Elaboración: 2014-05-19 | Versión del Software: 1.0 |
| Requisitos Implementados: R001-V1.0 R002-V1.0 R003-V1.0 R004-V1.0 | Requisitos Corregidos: N/A |
| Descripción: <p>Se realiza el diseño e implementación de los menús que contendrá el sistema; el registro, modificación y eliminación de los departamentos, carreras, coordinador de egresados y empresas.</p> <p>Para realizar el ingreso, modificación o eliminación de carreras se debe tomar en cuenta que es dependiente de los departamentos; es decir, para eliminar un departamento, este no debe tener carreras asignadas a ese departamento.</p> <p>En el ingreso de los datos de graduados se carga la información personal, y dirección, posterior se carga la información laboral si el graduado está trabajando, y si tiene historia laboral se carga ésta información:</p> <p>En base a las preguntas se carga la experiencia laboral y títulos adicionales si el graduado posee.</p> <p>Los reportes se obtienen en base a la información generada y guardada en la base de datos.</p> <p>El código perteneciente a la versión 1.0 se encontrará en el repositorio de las TIC's sede Latacunga en la carpeta asignada para el efecto.</p> | |
| <p>Sofía Gallardo</p>  <p>TÉCNICO RESPONSABLE DEL DESARROLLO</p> | |

VERSION DE SOFTWARE**Proyecto:**

Sistema de Control de Graduados y Graduas

Fecha Elaboración:

2014-05-19

Versión del Software:

2.0

Requisitos Implementados:

R001-V2.0

R002-V2.0

R003-V2.0

R004-V2.0

Requisitos Corregidos:

N/A

Descripción:

Se realiza el diseño e implementación de los menús que contendrá el sistema; el registro, modificación y eliminación de los graduados, domicilio, experiencia laboral, preguntas, título adicional.

Para realizar el ingreso, modificación o eliminación de graduados, se debe ingresar el usuario y password del graduado, la información que ingrese, modifique o elimine modificará la información de las tablas de domicilio, experiencia laboral, preguntas, título adicional.

El código perteneciente a la versión 2.0 se encontrará en el repositorio de las TIC's sede Latacunga en la carpeta asignada para el efecto.

Sofía Gallardo



TÉCNICO RESPONSABLE DEL DESARROLLO

| | |
|--|--------------------------------------|
| VERSION DE SOFTWARE | |
| Proyecto: Sistema de Control de Graduados y Graduas | |
| Fecha Elaboración: 2014-05-19 | Versión del Software: 3.0 |
| Requisitos Implementados: R001-V3.0 | Requisitos Corregidos: N/A |
| Descripción: Se realiza el diseño e implementación de los menús que contendrá el sistema; el registro, modificación y eliminación de la encuesta. Permite ingresar, modificar, eliminar datos que tiene la encuesta. El código perteneciente a la versión 3.0 se encontrará en el repositorio de las TIC's sede Latacunga en la carpeta asignada para el efecto. | |
| <p>Sofía Gallardo</p>  <p>TÉCNICO RESPONSABLE DEL DESARROLLO</p> | |

VERSION DE SOFTWARE**Proyecto:**

Sistema de Control de Graduados y Graduas

Fecha Elaboración:

2014-05-19

Versión del Software:

4.0

Requisitos Implementados:

R001-V4.0

Requisitos Corregidos:

N/A

Descripción:

Se realiza el diseño e implementación de los reportes asignados con su respectivo botón para exportar a Excel.

El código perteneciente a la versión 4.0 se encontrará en el repositorio de las TIC's sede Latacunga en la carpeta asignada para el efecto.

Sofía Gallardo



TÉCNICO RESPONSABLE DEL DESARROLLO

Anexo 5. Pruebas

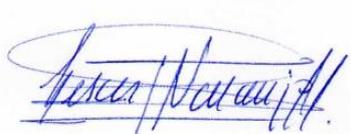
| PRUEBAS DE SISTEMA | | |
|---|--|--|
| Proyecto: Sistema de Control de Graduados y Graduadas | | |
| Fecha Elaboración: 2014-06-02 | Versión del Software: 1.0 | |
| Requisitos Validados: R001-V1.0 R002-V1.0 R003-V1.0 R004-V1.0 R005-V1.0 | Técnico Revisor: Sofía Gallardo | |
| Tiempo de Revisión: 4h | Tiempo de Soluciones: 5h | |
| Descripción de Fallos: Se ingresó una carrera pero al seleccionar el departamento no se mostraron todos los departamentos que esto afecta al ingreso de la carrera porque no se puede seleccionar la carrera específica que necesitamos, se cambiará la consulta en la base de datos para que se visualicen todos los departamentos del sistema y el usuario puede digitar y con el evento autocomplete muestre las posibles coincidencias. | | |
| No. | Fallo: | Solución: |
| 001-V1.0 | No se visualizan todas las carreras | Modificar el barrido de la consulta a la base de datos para que se visualicen todas las carreras |
| Sofía Gallardo  TÉCNICO RESPONSABLE DEL DESARROLLO | | |

| PRUEBAS DE SISTEMA | | |
|---|--|---|
| Proyecto: Sistema de Control de Graduados y Graduas | | |
| Fecha Elaboración: 2014-06-30 | Versión del Software: 2.0 | |
| Requisitos Validados: R001-V2.0 R002-V2.0 R003-V2.0 R004-V2.0 R005-V2.0 | Técnico Revisor: Sofía Gallardo | |
| Tiempo de Revisión: 4h | Tiempo de Soluciones: 5h | |
| Descripción de Fallos: Se ingresó los datos del graduado, y en el campo número de domicilio no permite ingresar letras. Al seleccionar otra opción de tipo de contrato no permite ingresar en el campo de texto la información. | | |
| No. | Fallo: | Solución: |
| 001-V2.0 | Campo domicilio no acepta letras | Modificar el ingreso para que reciba caracteres alfanuméricos |
| 002-V2.0 | Al seleccionar el combo de tipo de contrato otros no se puede ingresar datos en la caja de texto | Se valida el combo para habilitar la caja de texto |
| Sofía Gallardo | | |
|  | | |
| TÉCNICO RESPONSABLE DEL DESARROLLO | | |

| PRUEBAS DE SISTEMA | | |
|--|---|---|
| Proyecto: Sistema de Control de Graduados y Graduas | | |
| Fecha Elaboración: 2014-07-21 | Versión del Software: 3.0 | |
| Requisitos Validados: R001-V3.0 | Técnico Revisor: Sofía Gallardo | |
| Tiempo de Revisión: 4h | Tiempo de Soluciones: 5h | |
| Descripción de Fallos: Se ingresa las preguntas de la encuesta y se hace una prueba ingresando datos de un graduado para que salgan las preguntas, al ingresar la pregunta 7 se sale del aplicativo. | | |
| No. | Fallo: | Solución: |
| 001-V3.0 | Al ingresar los datos en la pregunta 7 se sale del aplicativo | Se validan los botones de aceptar, cancelar , salir |
| Sofía Gallardo  TÉCNICO RESPONSABLE DEL DESARROLLO | | |

| | | |
|---|---|--|
| PRUEBAS DE SISTEMA | | |
| Proyecto: Sistema de Control de Graduados y Graduas | | |
| Fecha Elaboración: 2014-08-15 | Versión del Software: 4.0 | |
| Requisitos Validados: R001-V4.0 | Técnico Revisor: Sofía Gallardo | |
| Tiempo de Revisión: 4h | Tiempo de Soluciones: 5h | |
| Descripción de Fallos: Al listar el reporte de Graduados por carrera no muestra los datos | | |
| No. | Fallo: | Solución: |
| 001-V1.0 | No se lee correctamente la información desde la base de datos | Se modifica el proceso de lectura de datos para que visualice en la pantalla esto se aplica a todos los reportes |
| Sofía Gallardo | | |
|  | | |
| TÉCNICO RESPONSABLE DEL DESARROLLO | | |

Anexo 6. Liberación

| ENTREGA DEL SISTEMA | |
|--|--|
| Proyecto: Sistema de Control de Graduados y Graduas | |
| Fecha Elaboración: 2015-05-20 | Versión del Software: 4.0 |
| Personal Capacitado: <hr style="width: 20%; margin: auto;"/> Ing. César Naranjo | Técnico Capacitador: Adela Sofía Gallardo Cueva |
| Novedades de Puesta en Marcha y la Capacitación: Se revisa con la persona a ser capacitada el funcionamiento del sistema. | |
| Detalle de la Versión: El perfil de acceso a usuarios maneja el perfil administrador el cual es el responsable de ingresar y mantener actualizadas las opciones de carrera, departamento y coordinador de carrera, mantenimiento a las preguntas der la encuesta y listar los reportes del sistema. Con la información que se encuentre ingresada en el sistema se listarán los siguientes reportes: <i>Reporte – Graduados por Carrera</i> <i>Reporte – Ficha de Graduado</i> <i>Reporte – Detalle de Asignatura Representativa</i> <i>Reporte – Detalle de Asignatura a Analizar</i> <i>Reporte – Detalle de Sueldos por Empresa y puesto</i> Estadísticas por: <i>Graduados por Carrera</i> <i>Información Personal</i> <i>Información Laboral</i> <i>Formación Recibida</i> <i>Formación Universitaria</i> | |
| Ing. César Naranjo  RESPONSABLE UNIDAD REQUIRENTE | Sofía Gallardo  RESPONSABLE DEL GRUPO DE DESARROLLO |



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por la señora a: **ADELA
SOFÍA GALLARDO CUEVA**

En la ciudad de Latacunga, a los **20 días del mes de mayo del 2015.**

Aprobado por:

Ing. Edgar Montaluisa

DIRECTOR DEL PROYECTO

Ing. Lucas Garcés

DIRECTOR DE CARRERA

Dr. Rodrigo Vaca

SECRETARIO ACADÉMICO