

**ESCUELA POLITÉCNICA DEL EJÉRCITO**  
**SEDE LATACUNGA**



**CARRERA DE INGENIERIA DE SISTEMAS E**  
**INFORMÁTICA**

**TEMA: "SISTEMA DE REGISTRO Y REPORTE DE ASISTENCIA DE BABEL SOFTWARE S.A. DESDE CUALQUIER ESTACIÓN DE TRABAJO MEDIANTE ACCESO REMOTO, UTILIZANDO TECNOLOGÍA MICROSOFT .NET 2005 Y REPORTING SERVICES."**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN**  
**SISTEMAS E INFORMÁTICA**

**DIANA CAROLINA REYES REYES**

**Latacunga, Julio 2009**

# ESCUELA POLITÉCNICA DEL EJÉRCITO

## CARRERA DE INGENIERIA DE SISTEMAS E INFORMÁTICA

### CERTIFICADO

ING. SANTIAGO JÁCOME (DIRECTOR)

ING. RAÚL CAJAS (CODIRECTOR)

### **CERTIFICAN:**

Que el trabajo titulado "SISTEMA DE REGISTRO Y REPORTE DE ASISTENCIA DE BABEL SOFTWARE S.A. DESDE CUALQUIER ESTACIÓN DE TRABAJO MEDIANTE ACCESO REMOTO, UTILIZANDO TECNOLOGÍA MICROSOFT .NET 2005 Y REPORTING SERVICES." realizado por la señorita: Diana Carolina Reyes Reyes ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la aplicación de conocimientos y al desarrollo profesional, **SI** recomiendan su publicación.

El mencionado trabajo consta de UN empastado y UN disco compacto el cual contiene los archivos en formato portátil de Acrobat. Autorizan a la señorita: Diana Carolina Reyes Reyes que lo entregue al ING. Edison Espinoza, en su calidad de Director de Carrera.

Latacunga, 22 de julio de 2009.

-----  
Ing. Santiago Jácome

**DIRECTOR**

-----  
Ing. Raúl Cajas

**CODIRECTOR**

**ESCUELA POLITÉCNICA DEL EJÉRCITO**  
**CARRERA DE INGENIERÍA DE SISTEMAS E INFORMATICA**

**AUTORIZACIÓN**

Yo, Diana Carolina Reyes Reyes

Autorizo a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la Institución del trabajo “SISTEMA DE REGISTRO Y REPORTE DE ASISTENCIA DE BABEL SOFTWARE S.A. DESDE CUALQUIER ESTACIÓN DE TRABAJO MEDIANTE ACCESO REMOTO, UTILIZANDO TECNOLOGÍA MICROSOFT .NET 2005 Y REPORTING SERVICES.” cuyo contenido, ideas y criterios son de mí exclusiva responsabilidad y autoría.

Latacunga, 4 de julio del 2009

-----  
Diana Carolina Reyes Reyes

C. I. No.-1718885609

## DEDICATORIA

Agradezco a Dios, por haberme otorgado vida, sabiduría e inteligencia para emprender y culminar con éxito todas las metas que me he propuesto.

A mí madre, por el esfuerzo, sacrificio, perseverancia, tenacidad y amor puesto en cada momento de su vida tanto para salir adelante como para educarnos y forjar nuestro carácter, dedico mi logro a ella con gratitud y cariño como muestra trascendental de su incondicional apoyo hacia mí.

A mí Amado Esposo, pilar fundamental de mi vida que desde la distancia ha permanecido a mi lado apoyándome y brindándome valor para alcanzar este objetivo.

A mi abuelita por permanecer junto a mí cuando más la necesitaba prodigándome siempre atenciones y cuidados.

A mi Director de Tesis, Ingeniero Santiago Jácome, por su guía, enseñanza y sabios aportes para con mi formación y culminación exitosa de mi carrera.

A mi Codirector de Tesis, Ingeniero Raúl Cajas, por su paciencia y dirección en el desarrollo de mi proyecto.

A mi Coordinador, Ingeniero Esteban Oviedo, por las oportunidades brindadas en su empresa tanto para salir adelante como adquirir nuevos conocimientos y experiencias en mi carrera.

A todos mis profesores de la ESPE, por sus valiosas enseñanzas y conocimientos impartidos durante mi vida académica.

A mis compañeros, Juanito y Cristóbal, por brindarme apoyo y soporte durante el tiempo que nos encontramos fuera de nuestro país.

*Dícarere*

# INDICE

<b>I. CAPÍTULO.....</b>	<b>- 6 -</b>
<b>ACERCA DE BABEL SOFTWARE S.A. ....</b>	<b>- 6 -</b>
<b>1.1 RESEÑA HISTÓRICA .....</b>	<b>- 6 -</b>
<b>1.2 MISIÓN .....</b>	<b>- 7 -</b>
<b>1.3 VISIÓN.....</b>	<b>- 7 -</b>
<b>1.4 DIVISIÓN DEPARTAMENTAL .....</b>	<b>- 8 -</b>
<b>1.5 GRUPOS DE TRABAJO .....</b>	<b>- 9 -</b>
<b>1.6 HORARIO DE TRABAJO.....</b>	<b>- 11 -</b>
<b>1.6.2 HORARIOS ESPECIALES .....</b>	<b>- 11 -</b>
<b>1.6.3 CUMPLIMIENTO DEL HORARIO .....</b>	<b>- 11 -</b>
<b>1.6.4 INASISTENCIAS AL TRABAJO .....</b>	<b>- 12 -</b>
<b>1.6.5 AUSENCIAS DE LAS INSTALACIONES.....</b>	<b>- 12 -</b>
<b>II. CAPITULO.....</b>	<b>- 13 -</b>
<b>METODO DE DESARROLLO DE SOFTWARE ITERATIVO INCREMENTAL UTILIZANDO UNA METODOLOGÍA RUP Y UML .....</b>	<b>- 13 -</b>
<b>2.1 INTRODUCCIÓN.....</b>	<b>- 13 -</b>
<b>2.2 LA ITERACIÓN Y EL INCREMENTO .....</b>	<b>- 13 -</b>
<b>2.2.1 LA ITERACIÓN.....</b>	<b>- 14 -</b>
<b>2.2.2 PLANIFICACIÓN Y SECUENCIACIÓN DE LAS ITERACIONES.....</b>	<b>- 14 -</b>
<b>2.2.3 EL INCREMENTO.....</b>	<b>- 15 -</b>
<b>2.3 EL PROCESO UNIFICADO DE DESARROLLO RUP .....</b>	<b>- 16 -</b>
<b>2.3.1 ORÍGENES .....</b>	<b>- 16 -</b>
<b>2.3.2 CARACTERÍSTICAS ESENCIALES .....</b>	<b>- 16 -</b>
<b>2.3.3 ESTRUCTURA DEL PROCESO RUP .....</b>	<b>- 19 -</b>
<b>2.3.4 PRÁCTICAS .....</b>	<b>- 22 -</b>
<b>2.4 LENGUAJE UNIFICADO DE MODELADO UML.....</b>	<b>- 24 -</b>
<b>2.4.1 INTRODUCCIÓN .....</b>	<b>- 25 -</b>
<b>2.4.2 ELEMENTOS DE UML .....</b>	<b>- 25 -</b>
<b>III. CAPÍTULO .....</b>	<b>- 35 -</b>
<b>HERRAMIENTAS DE DESARROLLO .....</b>	<b>- 35 -</b>

3.1 MICROSOFT VISUAL STUDIO 2005.....	- 35 -
3.1.1 INTRODUCCIÓN .....	- 35 -
3.1.2 DEFINICIÓN .....	- 37 -
3.1.3 VISUAL STUDIO 2005 PROFESSIONAL EDITION.....	- 37 -
3.1.4 VISUAL STUDIO 2005 STANDARD EDITION.....	- 38 -
3.1.5 NOVEDADES PRINCIPALES .....	- 39 -
3.1.6 MICROSOFT VISUAL STUDIO 2005 APPLICATION PLATFORM ...	- 41 -
3.2 REPORTING SERVICES 2005 .....	- 43 -
3.2.1 INTRODUCCIÓN .....	- 43 -
3.2.2 DEFINICIÓN .....	- 43 -
3.2.3 ARQUITECTURA DE REPORTING SERVICES .....	- 44 -
3.2.4 INSTALAR SQL SERVER REPORTING SERVICES.....	- 45 -
3.2.5 REPORT DEFINITION LANGUAGE RDL .....	- 46 -
3.2.6 DISEÑAR Y CREAR INFORMES .....	- 47 -
3.2.7 ADMINISTRAR REPORTING SERVICES.....	- 47 -
IV.    CAPITULO .....	- 50 -
DESARROLLO DEL SISTEMA REGASIS.....	- 50 -
4.1 CONCEPTUALIZACIÓN.....	- 50 -
4.1.1 NEGOCIACIÓN.....	- 50 -
4.1.1.1 ENTREVISTAS .....	- 50 -
4.1.1.2 DECLARACIÓN DE TRABAJO .....	- 57 -
4.1.2 ESPECIFICACIÓN DE REQUERIMIENTOS.....	- 65 -
4.2 PLANEACIÓN .....	- 85 -
4.2.1 PLAN DE TOMA DE REQUISITOS .....	- 85 -
4.2.2 PLAN DE IMPLANTACIÓN .....	- 90 -
4.3 EJECUCIÓN .....	- 95 -
4.3.1 ESPECIFICACIÓN DE DISEÑO .....	- 95 -
4.3.2 CASOS DE PRUEBA.....	- 150 -
4.4 CONTROL .....	- 172 -
4.4.1 INFORME DE RIESGOS.....	- 172 -
4.5 CIERRE.....	- 178 -
4.5.1 INFORME FINAL .....	- 178 -

<b>V. CAPÍTULO .....</b>	<b>- 184 -</b>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>- 184 -</b>
<b>5.1 CONCLUSIONES.....</b>	<b>- 184 -</b>
<b>5.2 RECOMENDACIONES .....</b>	<b>- 185 -</b>

## ÍNDICE DE FIGURAS

<b>Figura 1.1 Logo de Babel.....</b>	<b>- 6 -</b>
<b>Figura 1.2 Organigrama de Babel .....</b>	<b>- 8 -</b>
<b>Figura 1.3 Servidores de Babel.....</b>	<b>- 9 -</b>
<b>Figura 1.4 Mapa de Red de Babel .....</b>	<b>- 10 -</b>
<b>Figura 2.1 Una Iteración RUP .....</b>	<b>- 18 -</b>
<b>Figura 2.2 Proceso RUP .....</b>	<b>- 19 -</b>
<b>Figura 2.3 Diagrama de Casos de Uso .....</b>	<b>- 27 -</b>
<b>Figura 2.4 Diagrama de Clases .....</b>	<b>- 29 -</b>
<b>Figura 2.5 Diagrama de Secuencia .....</b>	<b>- 29 -</b>
<b>Figura 2.6 Diagrama de Colaboración .....</b>	<b>- 30 -</b>
<b>Figura 2.7 Diagrama de Estado.....</b>	<b>- 32 -</b>
<b>Figura 2.8 Diagrama de Actividad.....</b>	<b>- 33 -</b>
<b>Figura 2.9 Diagrama de Relación de Entidad.....</b>	<b>- 34 -</b>
<b>Figura 3.1 Arquitectura de Reporting Services .....</b>	<b>- 45 -</b>
<b>Figura 4.1 REGASIS .....</b>	<b>- 72 -</b>
<b>Figura 4.2 Arquitectura del Sistema .....</b>	<b>- 119 -</b>



## RESUMEN

El nacimiento de una necesidad tecnológica en la Empresa Babel Software S.A para el registro de entradas, salidas y ausencias del personal hace inevitable la razón de brindar una solución informática real a la misma, a través de un sistema que le permita realizar de forma remota y descentralizada el registro y reporte de asistencia de sus funcionarios, desde cualquier estación de trabajo, utilizando tecnología de punta como: Microsoft .Net 2005 y Reporting Services, las metodologías RUP y UML para el desarrollo del software y la aplicación de conceptos tales como: Aplicaciones Distribuidas, Programación en Capas, SOA, Web Services, Manejo de Threads, Procedimientos Almacenados, Burbujas de Información, entre otros.

El sistema REGASIS, es un sistema de desarrollo a la medida que fue realizado exclusivamente para funcionar en el ambiente de BABEL Software S.A.

Para el desarrollo de este proyecto de tesis, se propusieron 5 capítulos:

En el capítulo I, se habla de BABEL Software S.A. su historia, infraestructura, misión, visión, división departamental y organización interna, con el fin de conocer su necesidad, establecer parámetros y brindar una solución acorde a su contexto.

En el capítulo II, se realiza un análisis del Método RUP y UML basado en la Metodología de Desarrollo Iterativo Incremental, como guía para el desarrollo del sistema REGASIS.

En el capítulo III, se realiza una investigación de las herramientas a utilizarse en el desarrollo del Sistema REGASIS, Microsoft .NET 2005 y Reporting services, referentes a beneficios, funcionalidad, requerimientos, instalación y configuración.

En el capítulo IV, se hace referencia al desarrollo en sí del Sistema REGASIS aplicando el Método RUP y UML y las Herramientas Microsoft .NET 2005 y Reporting services, seguido del capítulo V donde constan las conclusiones y recomendaciones.

# I. CAPÍTULO

## ACERCA DE BABEL SOFTWARE S.A.



Figura 1.1 Logo de Babel

### 1.1 RESEÑA HISTÓRICA

Babel es una empresa familiar, que nació hace 6 años gracias al esfuerzo y sacrificio de dos hermanos costarricenses originarios de Pérez Zeledón. Su sueño de formar una empresa dedicada al análisis, diseño e implementación de soluciones informáticas se vio idealizada al fundar la empresa BABEL Software en Enero del 2002 con la visión de convertirse en la mejor opción para las organizaciones que deseen renovar o mejorar su plataforma de tecnologías de información, como medio para incrementar el rendimiento en la ejecución de los procesos.

Durante el lapso que la compañía lleva en el mercado, ha sabido aprovechar muy bien el auge de la industria de la programación, dedicándose a desarrollar soluciones basadas en software a la medida de sus clientes, pues elabora productos específicos a solicitud de empresas o bancos, adaptándolos a la plataforma que estos utilicen, lo que le ha permitido crecer en mas del 300% y evolucionar hasta constituirse en socia comercial del productor de software más importante del mundo, Microsoft y colocarse en la máxima categoría que otorga esta compañía a nivel mundial: Microsoft Gold Certified Partner, gracias a que cuenta con la mayor cantidad de colaboradores certificados. <sup>1</sup>

---

<sup>1</sup> <http://www.skyscraperlife.com/noticiascr/11068-babel-software-firma-costarricense-de-perez-zeledon-para-el-mundo-2.html>

Están inmersas en Babel, competencias tales como:

- Data Management Solutions
- Business Intelligence
- Information Worker Solutions
- Mobility Solutions
- Security Solutions
- SOA and Business Process
- Advanced Infrastructure Solutions
- Networking Infrastructure Solutions

Así Babel Software se ha consolidado tanto en el mercado nacional costarricense como en el mercado mundial, de allí que en el año 2008 se da la apertura de una nueva oficina en Ecuador con cerca de 20 ingenieros.

En este mismo año, se vuelve también aliada comercial de Oracle, otro grande de la industria a nivel mundial, con lo que logra la consolidación definitiva a nivel internacional y lo que le ayuda a fortalecer relaciones con sus actuales socios comerciales, con los que ha desarrollado hasta el día de hoy más de 200 proyectos, tanto en instituciones del sector público como en empresas del sector privado.

## **1.2 MISIÓN**

Ser un proveedor de soluciones informáticas que colaboren fuertemente en el incremento de la productividad de las empresas que hagan uso de nuestros productos y servicios, siendo a la vez una empresa sólida y rentable para sus accionistas y que brinde toda la confianza y seguridad que su persona espera.

## **1.3 VISIÓN**

Ser reconocida internacionalmente como una compañía consolidada y confiable, que brinda las mejores soluciones en el área de la Tecnología de Información.

1.4 DIVISIÓN DEPARTAMENTAL

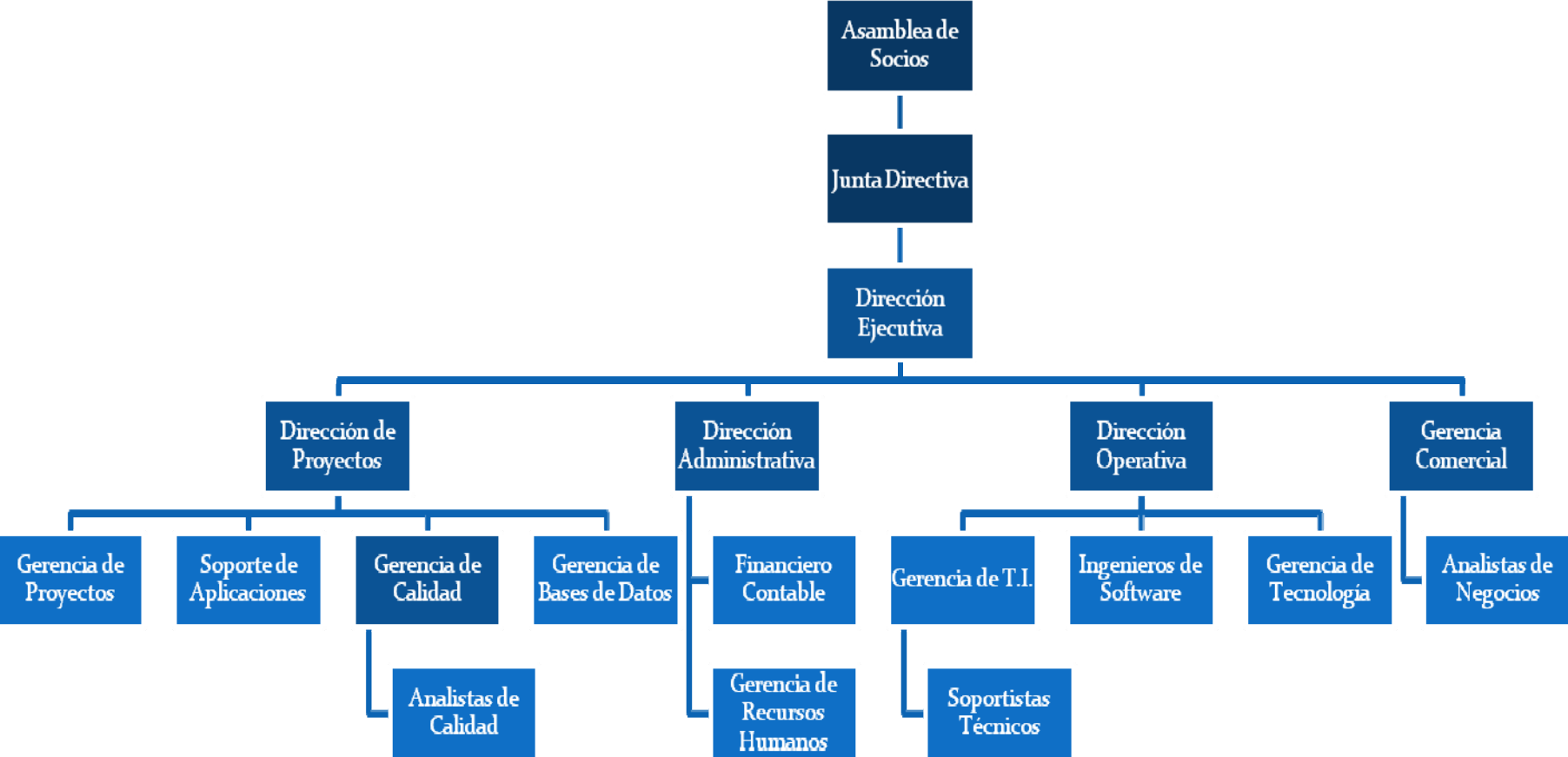
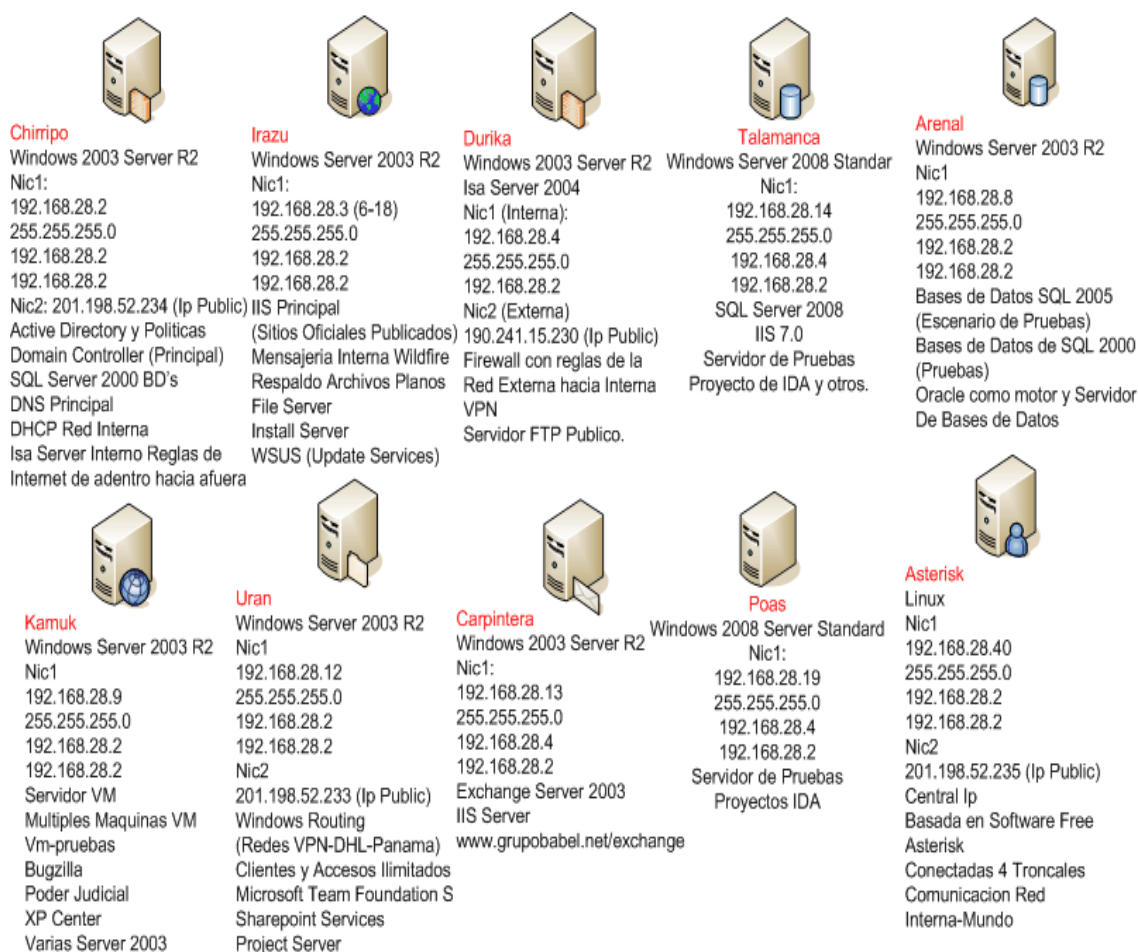


Figura 1.2 Organigrama de Babel

## 1.5 GRUPOS DE TRABAJO

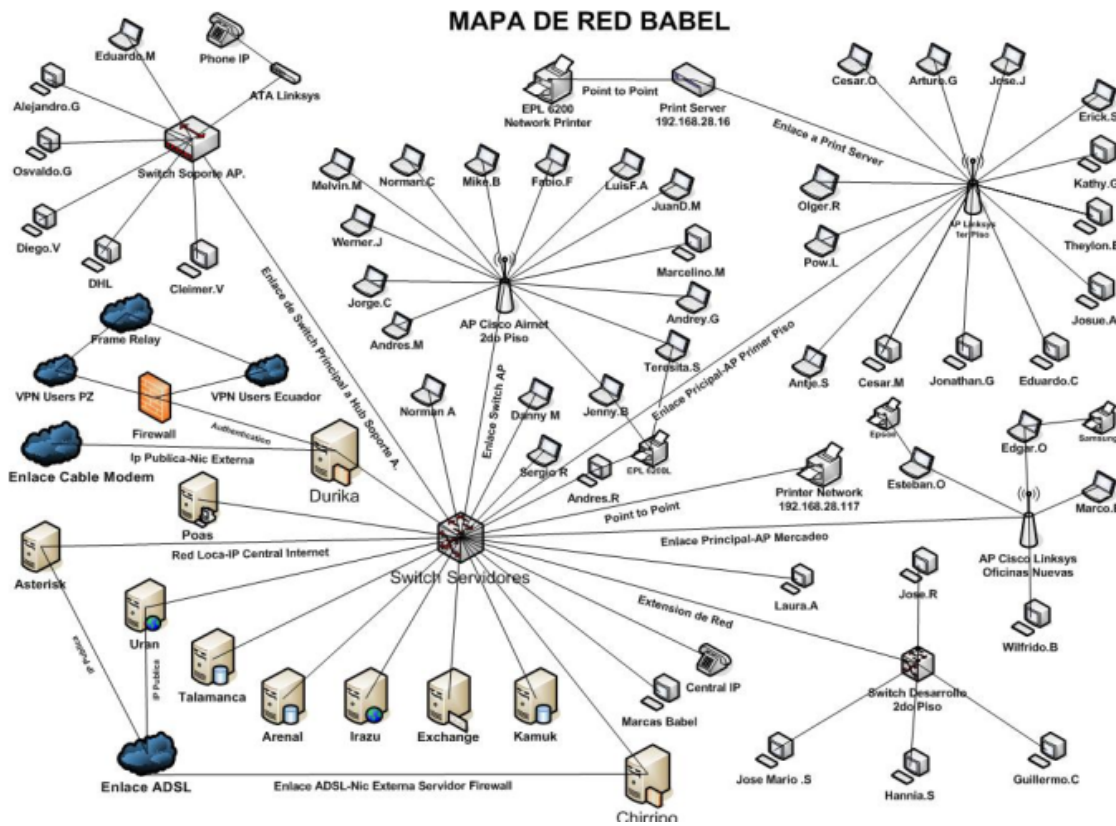
BABEL Software S.A. cuenta con varios servidores:



**Figura 1.3 Servidores de Babel**

La Base de Datos: BABELSYS, donde se encuentra almacenada la información de los empleados, las asistencias y otros, esta localizada en el servidor Arenal, mientras que las políticas y el Active Directory están en el servidor Chirripó y las publicaciones de los sitios tales como el Portal Interno de BABEL y más están en el servidor Arenal.

La red de BABEL está estructurada con una fusión de las topologías: Estrella y Malla, tal como se muestra en el siguiente gráfico:



**Figura 1.4 Mapa de Red de Babel**

Dentro de esta red los Grupos de Trabajo están organizados de la siguiente manera:

- Administrativos
- Base de Datos
- Comercial
- Desarrolladores
- Directores
- IT Technical Support
- Project Managers
- QA
- Recursos Humanos
- Soporte de Aplicaciones
- TFS
- VPN

## **1.6 HORARIO DE TRABAJO**

La jornada laboral completa de Babel Software S.A. consta de 48 horas por semana.

### **1.6.1 HORARIO REGULAR**

El horario regular de la empresa es:

- Lunes: 8:00 am – 5:45 pm
- Martes: 8:00 am – 5:45 pm
- Miércoles: 8:00 am – 5:45 pm
- Jueves: 8:00 am – 5:45 pm
- Viernes: 8:00 am – 5:00 pm

Todos los funcionarios deben estar disponibles siempre que se los requiera dentro de este horario, teniendo derecho a una hora de descanso al día durante este período, a utilizarse como la deseen, ya sea dentro o fuera de las instalaciones de la empresa, Aunque el funcionario deberá cumplir con este horario, para efectos de evaluación de su desempeño tendrá prioridad el seguimiento por objetivos que le realice constantemente su jefe inmediato o la misma gerencia.

### **1.6.2 HORARIOS ESPECIALES**

Únicamente por circunstancias que la Gerencia considere justificadas, se asignará un horario especial a alguno de los empleados o profesionales contratados para laborar en las instalaciones de la empresa. Dicho horario especial deberá estar especificado en un acuerdo firmado por ambas partes.

### **1.6.3 CUMPLIMIENTO DEL HORARIO**

Los empleados y profesionales contratados deberán cumplir el horario que se les asigne. Si algún día, por algún motivo el funcionario incumple el horario de modo que no cumpla con las horas que le correspondían, su jefe inmediato le indicará si debe reponer ese tiempo, así como el momento y la forma en que deberá hacerlo.

Incurrir en incumplimientos de horario frecuente podría implicar una sanción que será determinada, en su momento, por la gerencia.

#### **1.6.4 INASISTENCIAS AL TRABAJO**

Cada vez que el funcionario falte al trabajo, le serán rebajadas las horas del pago correspondiente a ese período (Salvo que exista previamente un acuerdo con la gerencia). Si un funcionario sabe de ante mano que deberá faltar en algún momento específico, deberá comunicarlo a la Gerencia con anticipación. Incurrir en faltas frecuentemente podría implicar una sanción que será determinada, en su momento, por la gerencia.

#### **1.6.5 AUSENCIAS DE LAS INSTALACIONES**

Cada vez que un funcionario requiera ausentarse de las instalaciones de la empresa por un período mayor a diez minutos, ya sea para algún asunto de la empresa o personal deberá comunicarlo a su jefe inmediato; no hacerlo implicará una sanción inmediata determinada por la gerencia. Incurrir en ausencias (sin la comunicación correspondiente) frecuentemente podría implicar una sanción que será determinada, en su momento, por la Gerencia.



## **II. CAPITULO**

# **METODO DE DESARROLLO DE SOFTWARE ITERATIVO INCREMENTAL UTILIZANDO UNA METODOLOGÍA RUP Y UML**

### **2.1 INTRODUCCIÓN**

Durante todo el proceso de desarrollo de software se producen versiones incrementales, que se acercan al producto terminado. El proceso RUP permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software mediante una gestión sistemática de riesgos, lleva a cabo el proceso de desarrollo práctico, brindando amplias guías, plantillas y ejemplos para todas las actividades críticas, se integra estrechamente con herramientas Rational, permitiendo a los equipos de desarrollo aprovechar todas las ventajas de las características de los productos Rational, el Lenguaje de Modelado Unificado (UML) y otras prácticas óptimas de la industria. El lenguaje Unificado de modelado UML es una notación que es el resultado de la evolución de las notaciones previas en ingeniería de software, toma los aspectos fuertes de tres metodologías anteriores: OMT, Booch y OOSE, su notación se fundamenta en principios de modelado a adoptarse del Proceso Unificado de Desarrollo para modelar las actividades de un proyecto, los diagramas a utilizar en las diferentes etapas del desarrollo de los sistemas de información, pueden variar dependiendo del tamaño y tipo de sistema, por lo que es necesario organizarlos según las fases del Proceso Unificado.

### **2.2 LA ITERACIÓN Y EL INCREMENTO**

El desarrollo iterativo e incremental es un enfoque para construir software donde el ciclo de vida está compuesto por varias iteraciones. Las iteraciones son pequeños proyectos compuestos de varias actividades cuyo objetivo es entregar una parte de un sistema parcialmente completo, probado, integrado y estable. Todo

el software es integrado en cada entrega de cada iteración hasta obtener el producto software completo en la última iteración. La definición de dicho enfoque se enmarca en lo qué es una iteración, en la planificación de la misma y en el concepto generado de incremento.

### **2.2.1 LA ITERACIÓN**

Una iteración es un mini proyecto que tiene como resultado una versión interna de cada uno de los artefactos que pueden ser generados en un proceso de desarrollo de software. Se puede visualizar como un flujo de trabajo compuesto por las actividades fundamentales del proceso (especificación, análisis, diseño, implementación, etc.) adoptado por un equipo de desarrollo para utilizar y producir los artefactos en los que se puede dividir una solución software.

El conjunto total e integrado de las iteraciones representará la versión externa del software, es decir, el producto final; para llegar a ello, cada una de las iteraciones dentro del ciclo de vida del software cumplen diferentes objetivos. Las primeras iteraciones se centran en la comprensión del problema y de la tecnología, se preocupan por producir el análisis del negocio, actividad o proceso que apoyará e integrará el producto. Luego se orientan las iteraciones a establecer las bases del diseño (arquitectura) que se robustecerá a medida que fluye el ciclo de vida. Las iteraciones siguientes se concentrarán en la construcción e integración de los componentes que se van generando producto de las mismas.

### **2.2.2 PLANIFICACIÓN Y SECUENCIACIÓN DE LAS ITERACIONES**

Un ciclo de vida iterativo requiere más planificación y comprensión que un modelo en cascada, donde toda su planificación se realiza al principio y contiene mayor incertidumbre y falta de veracidad para las fases siguientes en el proceso de desarrollo. Por el contrario, en el modelo iterativo no se planifica el proyecto durante el inicio, sólo se dan los primeros pasos; no se intenta llevar a cabo la construcción sin establecer las bases de la planificación en iteraciones previas. Cada planificación de una iteración tiene en cuenta los

resultados de las iteraciones precedentes, la selección de la especificación o funcionalidades a realizar y el estado actual del ciclo de vida del proceso, y termina con la preparación de la versión interna.

La evolución del software (proceso o producto) no sucede sin un plan que la preceda, la iteración en el desarrollo de software pretende ordenar cada flujo de trabajo para obtener un camino directo en el cuál las primeras iteraciones proporcionan la base de conocimiento para las siguientes. Estas iteraciones dan como resultado un mayor conocimiento de los requisitos, los problemas, los riesgos y el dominio de la solución, mientras que las siguientes dan como resultado una serie de incrementos aditivos que conforman finalmente la versión externa, el producto final para el cliente. La secuencia de iteraciones, que siempre avanza, representará el éxito de la aplicación del modelo, sin tener que volver nunca a los resultados de una iteración previa para corregir el modelo debido a algo descubierto en una iteración posterior. El objetivo principal de la planificación de las iteraciones es realizar una secuenciación del trabajo de manera que puedan desarrollarse primero las decisiones, especificaciones y funcionalidades más importantes.

### **2.2.3 EL INCREMENTO**

El incremento es el resultado de una iteración y está representado por la diferencia entre la versión interna dada por una iteración y la versión interna obtenida de la siguiente.

En un momento del ciclo iterativo (secuencia de iteraciones) algunos módulos o subsistemas estarán terminados e integrados, contendrán toda la funcionalidad requerida y plasmada en los requisitos, estarán implementados, probados y validados; otros sistemas estarán parcialmente terminados y otros sin haberse iniciado. Un incremento se puede ver entonces como la diferencia entre dos momentos determinados en el ciclo de vida del proceso. Se van construyendo incrementos de manera iterativa, dando forma a cada uno de los subsistemas que representan el sistema final que será visualizado cuando se realice la integración del último incremento.

## **2.3 EL PROCESO UNIFICADO DE DESARROLLO RUP**

El proceso unificado de desarrollo RUP (Rational Unified Process) es el representante más conocido del proceso iterativo incremental. Tanto su aplicación como enseñanza, es ampliamente difundida por organizaciones, empresas de desarrollo de software, universidades y expertos en prácticas y procesos de ingeniería de software en todo el mundo.

### **2.3.1 ORÍGENES**

En la historia de RUP el antecedente más importante se ubica en 1967 con la Metodología Ericsson (Ericsson Approach) elaborada por Ivar Jacobson, una aproximación del desarrollo basado en componentes, que introdujo el concepto de Casos de uso. Entre los años de 1987 a 1995 Jacobson fundó la compañía Objectory AB y lanza el proceso de desarrollo Objectory (abreviación de Object Factory), posteriormente en 1995 la renombrada compañía Rational Software Corporation adquiere a Objectory AB y entre 1995 y 1997 desarrolla el proceso ROP (Rational Objectory Process) a partir del Objectory 3.8 y del enfoque metodológico de Rational (Rational Approach) adoptando el lenguaje unificado UML como lenguaje de modelado. Desde ese entonces y a la cabeza de Grady Booch, Ivar Jacobson y James Rumbaugh, Rational Software desarrolló e incorporó diversos elementos para expandir su ROP, destacándose especialmente el conocido modelado del negocio. En junio del 1998 se lanza el RUP-Rational Unified Process, el conocido Proceso Unificado de Desarrollo propuesto por la Rational Corporation, hoy parte de IBM.

### **2.3.2 CARACTERÍSTICAS ESENCIALES**

El proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de uso, está centrado en la arquitectura, y es iterativo e incremental.

Proceso dirigido por Casos de uso

Los Casos de uso son una técnica de modelado para la captura y representación de requisitos que obliga a pensar en términos de importancia para el usuario y no sólo en términos de funciones. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor agregado. Los Casos de uso representan los requisitos funcionales del sistema. En RUP los casos de uso no son sólo una herramienta para especificar los requisitos del sistema; también guían su diseño, implementación y prueba. Los casos de uso constituyen un elemento integrador y una guía de trabajo, que permiten crear los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada caso de uso, haciendo que cada modelo esté sincronizando con cada uno de ellos.

#### Proceso centrado en la arquitectura

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados en el proceso (desarrolladores y usuarios) y una perspectiva más clara del sistema completo, necesaria para controlar el desarrollo.

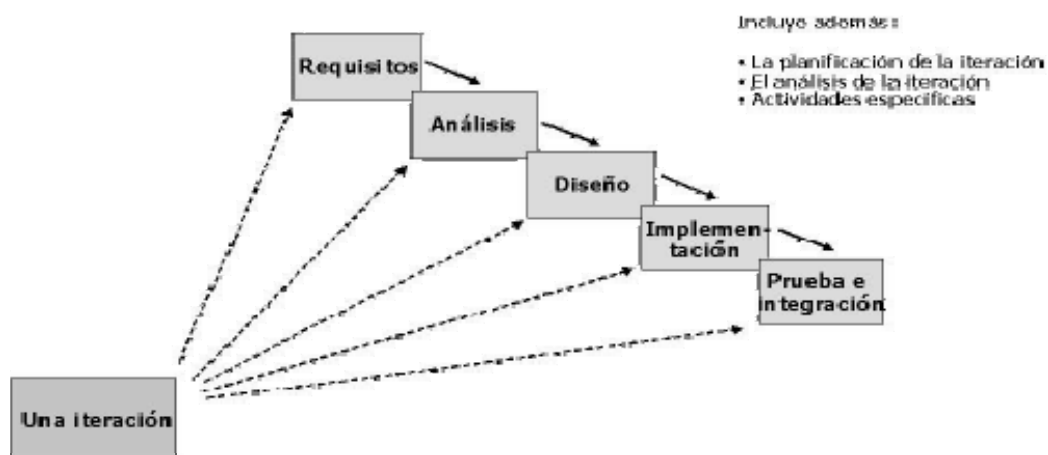
La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indica cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución, por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, el sistema operativo, los gestores de base de datos y protocolos, entre otros aspectos técnicos. Muchas de estas restricciones constituyen requisitos no funcionales del sistema. En el caso de RUP, además de utilizar los Casos de uso para guiar el proceso se presta especial atención al temprano establecimiento de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores presentados durante la construcción y el mantenimiento. Existe una interacción entre los Casos de uso y la arquitectura, los Casos de uso deben encajar en la arquitectura cuando se

llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de uso requeridos.

Un proceso iterativo e incremental.

En RUP se propone un proceso iterativo e incremental para equilibrar los casos de uso y la arquitectura, dividiendo este en partes más pequeñas o mini proyectos, cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo, actividades fundamentales o disciplinas) de la cual se obtiene un incremento que produce un crecimiento en el producto software.<sup>2</sup>

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina; se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso y toda la retroalimentación de la iteración pasada se utiliza para reajustar los objetivos de las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con una versión final del producto software.



**Figura 2.1 Una Iteración RUP**

<sup>2</sup> Fundamentos de la Metodología, 297224-RUP.pdf

### 2.3.3 ESTRUCTURA DEL PROCESO RUP

El proceso puede ser descrito en dos dimensiones o ejes: el horizontal representa la línea del tiempo y es considerado el eje de los aspectos dinámicos del proceso e indica las características del ciclo de vida del proceso, representado en fases (Inicio, Elaboración, Construcción y Transición), iteraciones e hitos y el vertical que representa los aspectos estáticos del proceso y lo describe en términos de componentes de proceso, disciplinas o flujos de trabajo “workflows”, artefactos, roles y actividades.

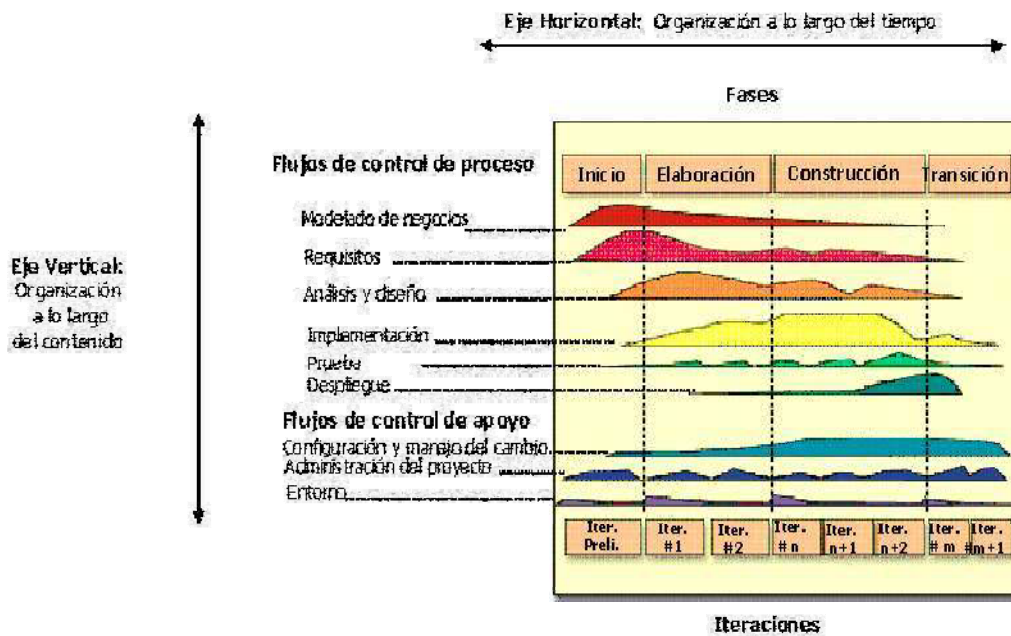


Figura 2.2 Proceso RUP

#### 2.3.3.1 Estructura Dinámica

RUP divide el proceso en cuatro fases:

- Inicio
- Elaboración
- Construcción
- Transición

Dentro de estas se realizan varias iteraciones en un número variable, las primeras iteraciones de las fases de Inicio y Elaboración se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una línea base para la arquitectura, la fase de Construcción se lleva a cabo por medio de una serie de iteraciones, para cada iteración se seleccionan algunos casos de uso, se refina su análisis y diseño, y se procede a su implementación y pruebas, se realiza una pequeña cascada para cada ciclo, se realizan tantas iteraciones como se requieran, hasta que se termine la implementación de una versión del producto y en la fase de Transición se pretende garantizar que se tiene un producto preparado para su entrega a los usuarios.

### **2.3.3.2 Estructura Estática**

Un proceso de desarrollo de software define quién hace, qué, cómo y cuándo. RUP define cuatro elementos: los roles, las actividades, los productos (artefactos) y los flujos de trabajo o disciplinas que responden a cada una de estas preguntas.

#### **Roles y Actividades**

Una persona puede desempeñar diversos roles, así como un mismo rol puede ser representado por varias personas. Las responsabilidades de un rol son tanto el llevar a cabo un conjunto de actividades como el ser el dueño de un conjunto de artefactos. Una actividad en concreto es una unidad de trabajo que una persona desempeña. RUP define grupos de roles, agrupados por participación en actividades relacionadas. Estos grupos son:

- Analistas: Analista de procesos de negocio, Diseñador del negocio, Analista de sistema y Especificador de requisitos.
- Desarrolladores: Arquitecto de software, Diseñador, Diseñador de interfaz de usuario, Diseñador de base de datos, Implementador e Integrador.



- Gestores: Jefe de proyecto, Jefe de control de cambios, Jefe de configuración, Jefe de pruebas, Jefe de despliegue, Ingeniero de procesos, Revisor de gestión del proyecto y Gestor de pruebas.
- Apoyo: Documentador técnico, Administrador de sistema, Especialista en herramientas, Desarrollador de cursos y Artista gráfico.
- Especialista en pruebas: Especialista en Pruebas (Tester), Analista de pruebas, Diseñador de pruebas.
- Otros roles: Stakeholders (participantes), Revisor, Coordinador de revisiones y Revisor técnico

## Artefactos

Los productos o artefactos son los resultados tangibles del proyecto, las cosas que se van creando, modificando y usando hasta obtener el producto final durante el proceso de desarrollo de software. Un artefacto puede ser cualquiera de los siguientes: un documento, un modelo (como el de casos de uso), un elemento propio del modelo (una clase o un caso de uso), un subsistema o componente.

## Flujos de trabajo (Disciplinas)

Con simplemente la enumeración de roles, actividades y artefactos no se define un proceso, se necesita contar con una secuencia de actividades realizadas por los diferentes roles, así como la relación entre los mismos. Un flujo de trabajo es una relación de actividades que producen unos resultados observables, y en RUP son:

- Modelado del negocio. Con esta disciplina se pretende llegar a un mejor entendimiento de la organización donde se va a implantar el producto (software).
- Requisitos. Esta es una de las disciplinas más importantes, porque en él se establece qué tiene que hacer exactamente el sistema a construir.
- Análisis y Diseño. El objetivo de este flujo es traducir los requisitos a una especificación que describe cómo implementar el sistema, mediante

su representación a través de los diferentes modelos UML definidos para las actividades de análisis y diseño.

- Implementación. En éste flujo se implementan las clases y objetos, en código fuente, binarios y ejecutables. Además se deben hacer las pruebas de unidad: cada desarrollador es responsable de probar las unidades que produzca. El resultado final de este flujo de trabajo es un sistema ejecutable.
- Pruebas. Este flujo de trabajo es el encargado de evaluar la calidad del producto que está siendo desarrollado, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino para ir integrándolo en todo el ciclo de vida.
- Despliegue. El objetivo de este flujo de trabajo es producir con éxito entregables del producto y distribuirlos a los usuarios.
- Gestión del proyecto. La gestión del proyecto es el arte de lograr un balance al administrar objetivos, riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.
- Configuración y control de cambios. El objetivo de ésta disciplina es mantener la integridad de todos los artefactos que se crean en el proceso, así como de mantener información del proceso evolutivo que se ha seguido.
- Entorno. La finalidad de éste flujo de trabajo es dar soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir el instante concreto del proceso en el que han de ser empleados.

#### **2.3.4 PRÁCTICAS**

En RUP se identifican seis prácticas con las que define una forma efectiva de trabajar en equipos de desarrollo de software.

Gestión de requisitos

RUP brinda una guía para encontrar, organizar, documentar, y seguir los cambios de los requisitos funcionales y sus restricciones. Utiliza la notación de Caso de Uso y escenarios para representar los requisitos.

#### Desarrollo de software iterativo

Desarrollo del producto mediante iteraciones con hitos bien definidos, en las cuales se repiten las actividades pero con distinto énfasis, según la fase del proyecto.

#### Desarrollo basado en componentes

La creación intensiva de sistemas en software requiere dividir cada sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica es un proceso de desarrollo que permite que el sistema se vaya creando a medida que se obtienen o se desarrollan sus componentes.

#### Modelado visual (usando UML)

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema software. Utilizar herramientas de modelado visual facilita la gestión de dichos modelos, permitiendo ocultar o exponer detalles cuando sea necesario. El modelado visual también ayuda a mantener la consistencia entre los artefactos del sistema: requisitos, diseños e implementaciones. El modelado visual ayuda a mejorar la capacidad del equipo para gestionar la complejidad del software.

#### Verificación continua de la calidad

Es importante que la calidad de todos los artefactos se evalúe varias veces durante el proceso de desarrollo, especialmente al final de cada iteración. En esta verificación las pruebas juegan un papel fundamental y se integran a lo largo de

todo el proceso. Para todos los artefactos, no necesariamente ejecutables, las revisiones e inspecciones deben ser continuas.

## Gestión de los cambios

El cambio es un factor de riesgo crítico en los proyectos de software. Los artefactos software cambian no sólo por acciones de mantenimiento posteriores a la entrega del producto, sino que durante el proceso de desarrollo aparecen nuevos cambios en los requisitos. Otro gran desafío que debe abordarse en la construcción de software es la participación de múltiples desarrolladores trabajando a la vez en un mismo incremento y quizás en distintas plataformas. La ausencia de una disciplina, rápidamente conduciría al caos. Las prácticas de Gestión de Cambios y de Configuración es la disciplina que RUP encarga para este aspecto.

## 2.4 LENGUAJE UNIFICADO DE MODELADO UML

La notación UML se deriva y unifica las tres metodologías de análisis y diseño orientada a objetos más extendidas<sup>3</sup>:

- Metodología de Grady Booch para la descripción de conjuntos de objetos y sus relaciones.
- Técnica de modelado orientada a objetos de James Rumbaugh (OMT: Object-Modeling Technique).
- Aproximación de Ivar Jacobson (OOSE: Object- Oriented Software Engineering) mediante la metodología de casos de uso (use case).

UML combina notaciones provenientes desde:

- Modelado Orientado a Objetos.
- Modelado de Datos–Modelado de Componentes.
- Modelado de Flujos de Trabajo (Workflows)

---

<sup>3</sup> [http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado)

El desarrollo de UML comenzó a finales de 1994 cuando Grady Booch y Jim Rumbaugh de Rational Software Corporation empezaron a unificar sus métodos. A finales de 1995, Ivar Jacobson y su compañía Objectory se incorporaron a Rational en su unificación, aportando el método OOSE.

De las tres metodologías de partida, las de Booch y Rumbaugh pueden ser descritas como centradas en objetos, ya que sus aproximaciones se enfocan hacia el modelado de los objetos que componen el sistema, su relación y colaboración. Por otro lado, la metodología de Jacobson es más centrada a usuario, ya que todo en su método se deriva de los escenarios de uso. UML se ha ido fomentando y aceptando como estándar desde el OMG, que es también el origen de CORBA, el estándar líder en la industria para la programación de objetos distribuidos. En 1997 UML 1.1 fue aprobada por la OMG convirtiéndose en la notación estándar de facto para el análisis y el diseño orientado a objetos.

UML es el primer método en publicar un meta-modelo en su propia notación, incluyendo la notación para la mayoría de la información de requisitos, análisis y diseño. Se trata pues de un meta-modelo auto-referencial (cualquier lenguaje de modelado de propósito general debería ser capaz de modelarse a sí mismo).

#### **2.4.1 INTRODUCCIÓN**

El lenguaje unificado de modelado (UML) sirve para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. UML no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente le ayuda a visualizar el diseño y a hacerlo más accesible para otros. UML está controlado por el grupo de administración de objetos (OMG) y es el estándar de descripción de esquemas de software.

UML está diseñado para su uso con software orientado a objetos, y tiene un uso limitado en otro tipo de cuestiones de programación.

#### **2.4.2 ELEMENTOS DE UML**

UML se compone de muchos elementos de esquematización que representan las diferentes partes de un sistema de software. Los elementos UML se utilizan para crear diagramas, que representa alguna parte o punto de vista del sistema.

### **2.4.2.1 Diagrama de Casos de Uso**

Los diagramas de casos de uso describen las relaciones y las dependencias entre un grupo de casos de uso y los actores participantes en el proceso.

Es importante resaltar que los diagramas de casos de uso no están pensados para representar el diseño y no puede describir los elementos internos de un sistema. Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema, y con el cliente, y resultan especialmente útiles para determinar las características necesarias que tendrá el sistema. En otras palabras, los diagramas de casos de uso describen qué es lo que debe hacer el sistema, pero no cómo.

#### Caso de uso

Los casos de uso describen las interacciones típicas entre los usuarios de un sistema y el sistema mismo. Representan la interfaz externa del sistema y especifican qué requisitos de funcionamiento debe tener este.

#### Reglas

- Cada caso de uso está relacionado como mínimo con un actor.
- Cada caso de uso es un iniciador (es decir, un actor)
- Cada caso de uso lleva a un resultado relevante (un resultado con “valor intrínseco”)
- Los casos de uso pueden tener relaciones con otros casos de uso.

<<include>> que especifica una situación en la que un caso de uso tiene lugar dentro de otro caso de uso

<<extends>> que especifica que en ciertas situaciones, o en algún punto (llamado punto de extensión) un caso de uso será extendido por otro.

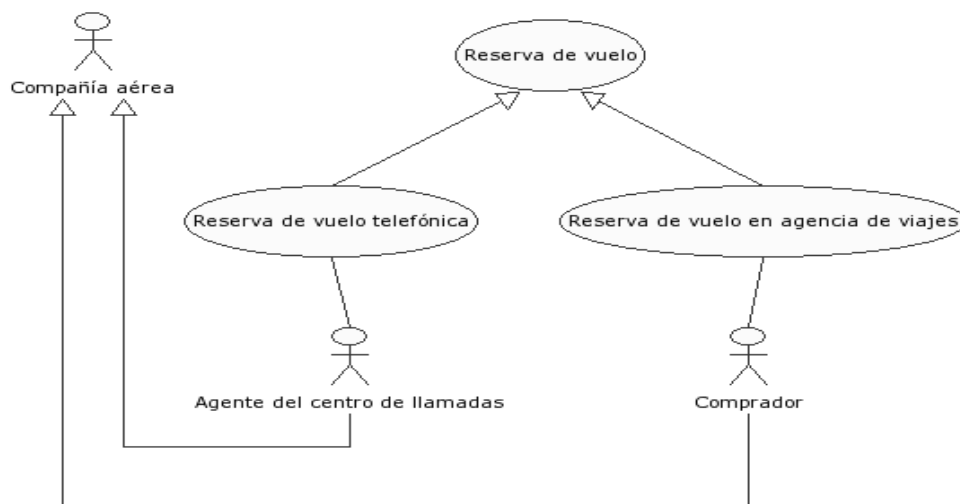
Generalización que especifica que un caso de uso hereda las características del “super” caso de uso, y puede volver a especificar algunas o todas ellas de una forma muy similar a las herencias entre clases.

## Actor

Un actor es una entidad externa (de fuera del sistema) que interacciona con el sistema participando (y normalmente iniciando) en un caso de uso. Los actores pueden ser gente real (por ejemplo, usuarios del sistema), otros ordenadores o eventos externos. Los actores no representan a personas físicas o a sistemas, sino su papel.

## Descripción de casos de uso

Las descripciones de casos de uso son reseñas textuales del caso de uso. Normalmente tienen el formato de una nota o un documento relacionado de alguna manera con el caso de uso, y explica los procesos o actividades que tienen lugar en el caso de uso.



**Figura 2.3 Diagrama de Casos de Uso**

### 2.4.2.2 Diagrama de Clases

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Los diagramas de clases son diagramas “estáticos” porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases “conocen” a qué otras clases o qué clases “son parte” de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

#### Clase

Una clase define los atributos y los métodos de una serie de objetos. Todos los objetos de esta clase (instancias de esa clase) tienen el mismo comportamiento y el mismo conjunto de atributos (cada objeto tiene el suyo propio). Las clases están representadas por rectángulos, con el nombre de la clase, atributos y operaciones de la clase en otros dos “compartimentos” dentro del rectángulo.

#### Atributos

En UML, los atributos se muestran al menos con su nombre, y también pueden mostrar su tipo, valor inicial y otras propiedades. Los atributos también pueden ser mostrados visualmente:

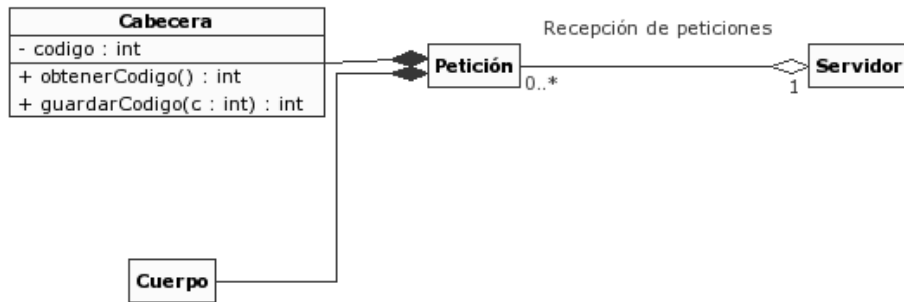
#### Operaciones

Las operaciones (métodos) también se muestran al menos con su nombre, y pueden mostrar sus parámetros y valores de retorno.

#### Asociaciones de clases

Las clases se pueden relacionar (estar asociadas) con otras de diferentes maneras.



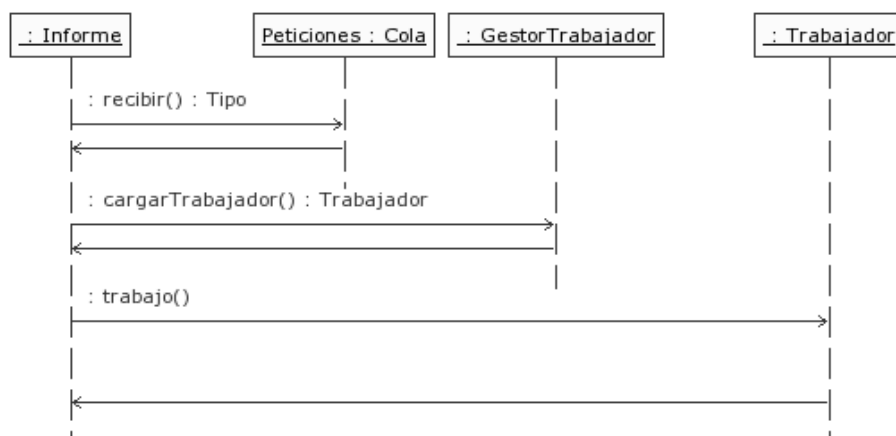


**Figura 2.4 Diagrama de Clases**

### 2.4.2.3 Diagramas de Secuencia

Los diagramas de secuencia muestran el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.

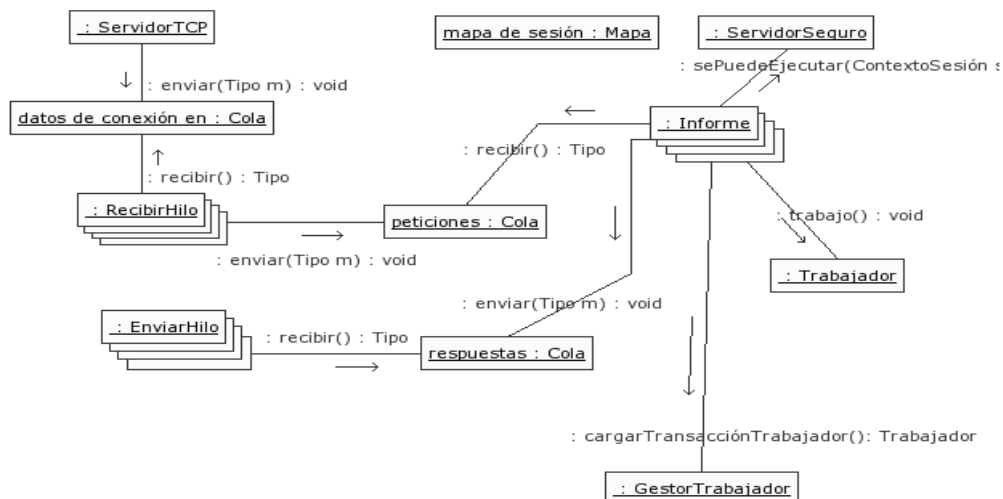


**Figura 2.5 Diagrama de Secuencia**

### 2.4.2.4 Diagramas de Colaboración

Los diagramas de colaboración muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esta es más o menos la misma información que la mostrada por los diagramas de secuencia, pero destacando la forma en que las operaciones se producen en el tiempo, mientras que los diagramas de colaboración fijan el interés en las relaciones entre los objetos y su topología.

En los diagramas de colaboración los mensajes enviados de un objeto a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje. Los diagramas de colaboración están indicados para mostrar una situación o flujo programa específicos y son unos de los mejores tipos de diagramas para demostrar o explicar rápidamente un proceso dentro de la lógica del programa.



**Figura 2.6 Diagrama de Colaboración**

#### 2.4.2.5 Diagrama de Estado

Los diagramas de estado muestran los diferentes estados de un objeto durante su vida, y los estímulos que provocan los cambios de estado en un objeto. Puede tener durante su vida uno de los siguientes estados:

- Listo
- Escuchando

- Trabajando
- Detenido

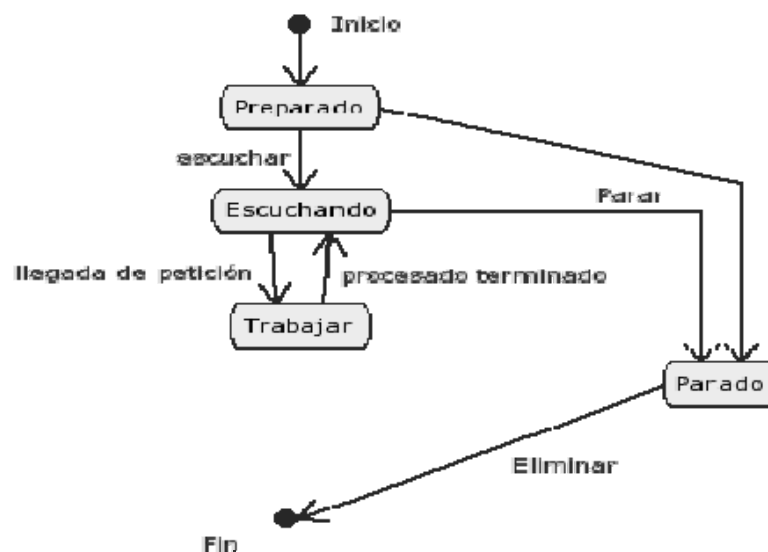
Los eventos que producen los cambios de estado son:

- Se crea el objeto
- El objeto recibe un mensaje de escucha
- Un cliente solicita una conexión a través de la red
- Un cliente finaliza una solicitud
- La solicitud se ejecuta y se termina
- El objeto recibe un mensaje de detención, entre otros.

## Estado

Los estados son los ladrillos de los diagramas de estado. Un estado pertenece a exactamente una clase y representa un resumen de los valores y atributos que puede tener la clase. Un estado UML describe el estado interno de un objeto de una clase particular

Hay dos tipos especiales de estados: inicio y fin. Son especiales en el sentido de que no hay ningún evento que pueda devolver a un objeto a su estado de inicio, y de la misma forma no hay ningún evento que pueda sacar a un objeto de su estado de fin.



## Figura 2.7 Diagrama de Estado

### 2.4.2.6 Diagrama de Actividad

Los diagramas de actividad describen la secuencia de las actividades en un sistema. Los diagramas de actividad son una forma especial de los diagramas de estado, que únicamente (o mayormente) contienen actividades.

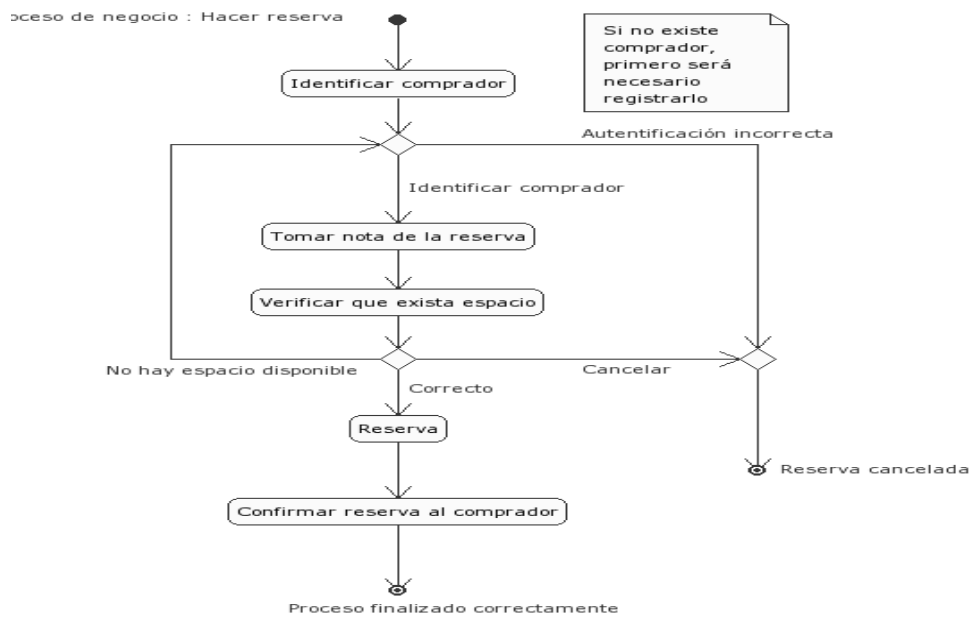
Los diagramas de actividad son similares a los diagramas de flujo procesales, con la diferencia de que todas las actividades están claramente unidas a objetos.

Los diagramas de actividad siempre están asociados a una clase, a una operación o a un caso de uso.

Los diagramas de actividad soportan actividades tanto secuenciales como paralelas. La ejecución paralela se representa por medio de iconos de fork/espera, y en el caso de las actividades paralelas, no importa en qué orden sean invocadas (pueden ser ejecutadas simultáneamente o una detrás de otra).

#### Actividad

Una actividad es un único paso de un proceso. Las actividades pueden formar jerarquías, lo que significa que una actividad puede estar formada de varias actividades “de detalle”, en cuyo caso las transiciones entrantes y salientes deberían coincidir con las del diagrama de detalle.



**Figura 2.8 Diagrama de Actividad**

#### 2.4.2.7 Elementos de Ayuda

Existen unos elementos en UML que no tiene un valor semántico real en la maqueta, pero que ayudan a clarificar partes del programa. Estos elementos son:

- Línea de texto
- Notas de texto y enlaces
- Cajas

Las líneas de texto son útiles para añadir información textual a un diagrama.

Las notas son útiles para añadir información más detallada de un objeto o una situación específica. Tienen la gran ventaja de que se pueden anclar a los elementos UML para mostrar que una nota “pertenece” a un objeto o situación específicos

Las cajas son rectángulos repartidos libremente que pueden usarse para juntar objetos haciendo los diagramas más legibles.

#### 2.4.2.8 Diagrama de Componentes

Los diagramas de componentes muestran los componentes del software y los artilugios de que está compuesto como los archivos de código fuente, las librerías o las tablas de una base de datos.

Los componentes pueden tener interfaces (es decir clases abstractas con operaciones) que permiten asociaciones entre componentes.

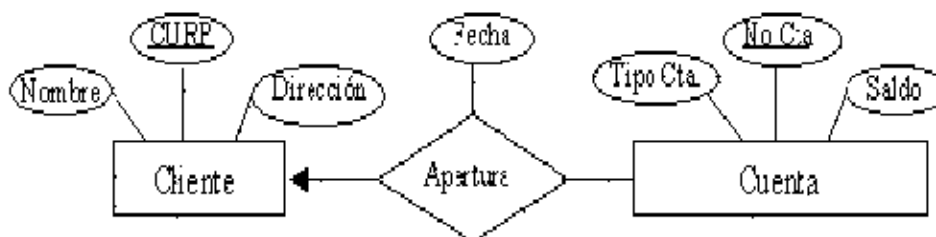
#### 2.4.2.9 Diagramas de Implementación

Los diagramas de implementación muestran las instancias existentes al ejecutarse así como sus relaciones. También se representan los nodos que identifican recursos físicos, típicamente un ordenador así como interfaces y objetos (instancias de las clases).

#### 2.4.2.10 Diagramas de Relación de Entidad

Este modelo representa a la realidad a través de un esquema gráfico empleando los terminología de entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver con el diagramado y se distinguen de otros por sus características particulares denominadas atributos, el enlace que rige la unión de las entidades esta representada por la relación del modelo.

En el diagrama un rectángulo representa a las entidades; una elipse a los atributos de las entidades, y una etiqueta dentro de un rombo indica la relación que existe entre las entidades, destacando con líneas las uniones de estas.



**Figura 2.9 Diagrama de Relación de Entidad**

## III. CAPÍTULO

### HERRAMIENTAS DE DESARROLLO

#### 3.1 MICROSOFT VISUAL STUDIO 2005

Visual Studio 2005 se empezó a comercializar a través de Internet a partir del 4 de Octubre de 2005, llegó a los comercios a finales del mes de Octubre en idioma inglés y en castellano el 4 de Febrero de 2006. La actualización más importante que recibieron los lenguajes de programación fue la inclusión de tipos genéricos, similares en muchos aspectos a las plantillas de C#. Con esto se consigue encontrar muchos más errores en la compilación en vez de en tiempo de ejecución, incitando a usar comprobaciones estrictas en áreas donde antes no era posible. C++ tiene una actualización similar con la adición de C++/CLI como sustituto de C# manejado.<sup>4</sup>

Se incluye un diseñador de implantación, que permite que el diseño de la aplicación sea validado antes de su implantación. También se incluye un entorno para publicación web y pruebas de carga para comprobar el rendimiento de los programas bajo varias condiciones de carga.

Visual Studio 2005 tiene varias ediciones radicalmente distintas entre sí: Express, Standard, Professional, Tools for Office, y 5 ediciones Visual Studio Team System.

##### 3.1.1 INTRODUCCIÓN

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones Web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C++, Visual C# y Visual J# utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo,

---

<sup>4</sup> <http://190.40.223.217/biblioteca/Informatica/Programacion/VisualNet.aspx>

dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y Servicios Web XML.

#### Visual Studio Tools para Office

Microsoft Visual Studio 2005 Tools para Microsoft Office System puede ayudarle a crear soluciones al extender documentos de Word 2003 y libros de Excel 2003 mediante Visual Basic y Visual C#.

#### Visual Web Developer

Visual Studio incluye un nuevo diseñador de páginas Web denominado Visual Web Developer que incluye muchas mejoras para la creación y edición de páginas Web ASP.NET y páginas HTML. Proporciona una forma más fácil y rápida de crear páginas de formularios Web Forms que en Visual Studio .NET 2003.

#### Aplicaciones para Dispositivos Inteligentes

El entorno integrado de Visual Studio incluye herramientas destinadas a dispositivos como los PDA y Smartphone.

#### Servicios Web XML

Los Servicios Web XML son aplicaciones que pueden recibir solicitudes y datos mediante XML a través de http, se puede obtener acceso a ellos mediante cualquier lenguaje, modelo de componente o sistema operativo.

#### Compatibilidad con XML

El Lenguaje de marcado extensible (XML) proporciona un método para describir datos estructurados. XML es un subconjunto de SGML optimizado para la entrega a través de Web. El Consorcio World Wide Web (W3C) define los estándares de XML para que los datos estructurados sean uniformes e independientes de las



aplicaciones. Visual Studio es totalmente compatible con código XML e incluye el Diseñador XML para facilitar la edición de XML y la creación de esquemas XML.

### Visual Studio Team System

Visual Studio 2005 Team System es una plataforma de herramientas del ciclo de vida del desarrollo de software extensible, integrado y productivo que ayuda a los equipos de desarrollo de software mediante la mejora de las comunicaciones y la colaboración durante todo el proceso de desarrollo.

### El entorno .NET Framework

Punto NET Framework es un entorno multilenguaje que permite generar, implantar y ejecutar aplicaciones y Servicios Web XML, consta de tres partes principales: Common Language Runtime, Clases de programación unificadas y ASP.NET.

### **3.1.2 DEFINICIÓN**

Microsoft Visual Studio 2005 es una completa herramienta de desarrollo de software que sirve para crear e implementar rápidamente una extensa gama de aplicaciones de Windows, web y dispositivos móviles.

### **3.1.3 VISUAL STUDIO 2005 PROFESSIONAL EDITION**

Visual Studio 2005 Professional Edition, permite la construcción de aplicaciones multicapa para la Web, dispositivos móviles, clientes inteligentes, o aplicaciones basadas en Microsoft Office

Con esta herramienta los desarrolladores profesionales pueden: disfrutar de un entorno de desarrollo altamente productivo con diseñadores visuales, lenguajes de programación y editores de código mejorados, desarrollar y depurar aplicaciones multicapa de servidor desde un mismo entorno unificado de desarrollo (Integrated Development Environment - IDE), construir soluciones para SQL Server 2005 utilizando herramientas visuales integradas de diseño de bases de datos e informes y crear sus propias herramientas que extiendan el IDE de Visual Studio usando el

SDK de Visual Studio, además de tener un acceso sin trabas al .NET Framework 2.0, el entorno robusto y plenamente funcional de la plataforma Microsoft.

### **3.1.3.1 Requisitos del Sistema para Visual Studio 2005 Professional Edition**

Para instalar Visual Studio 2005 Professional Edition se debe tener como requerimientos mínimos:

- Ordenador a 600 MHz
- Microsoft® Windows® 2000 Service Pack 4,  
Microsoft® Windows® XP Service Pack 2  
Microsoft® Windows® XP Professional x64  
Microsoft® Windows Server™ 2003 Service Pack 1  
Microsoft® Windows Server™ 2003, x64  
Microsoft® Windows Server™ 2003 R2  
Microsoft® Windows Server™ 2003 R2, x64  
Microsoft® Windows Vista™
- 192 MB de RAM
- GB de espacio en disco
- Unidad DVD-ROM
- Monitor de 1024x768 o superior con 256 colores
- Teclado y Microsoft Mouse o compatible

### **3.1.4 VISUAL STUDIO 2005 STANDARD EDITION**

Visual Studio 2005 Standard Edition es una herramienta sencilla pero muy poderosa de desarrollo para introducirse en la programación, la misma que permite a los desarrolladores: crear aplicaciones de línea de negocio usando Visual Basic, C#, C++ y J#, construir aplicaciones para Windows, la Web y dispositivos móviles todo desde el mismo entorno unificado de desarrollo y desarrollar aplicaciones cliente/servidor usando servicios Web y herramientas integradas de diseño para acceder a datos remotos

### 3.1.4.1 Requisitos del Sistema para Visual Studio 2005 Standard Edition

Para instalar Visual Studio 2005 Standard Edition se debe tener como requerimientos mínimos:

- Ordenador a 600 MHz
- Microsoft® Windows® 2000 Service Pack 4,  
Microsoft® Windows® XP Service Pack 2  
Microsoft® Windows® XP Professional x64  
Microsoft® Windows Server™ 2003 Service Pack 1  
Microsoft® Windows Server™ 2003, x64  
Microsoft® Windows Server™ 2003 R2  
Microsoft® Windows Server™ 2003 R2, x64  
Microsoft® Windows Vista™
- 192 MB de RAM
- GB de espacio en disco
- Unidad DVD-ROM
- Monitor de 1024x768 o superior con 256 colores
- Teclado y Microsoft Mouse o compatible

### 3.1.5 NOVEDADES PRINCIPALES

Visual Studio 2005 presenta novedades tanto dentro del propio entorno de desarrollo, como en el .NET Framework y el lenguaje de programación.

#### 3.1.5.1 Novedades Principales del Entorno de Desarrollo

- Soporte para refactoring: Refactorizar es el proceso de modificar el código fuente pero sin modificar su temática. El objetivo es adherirse al *best practices* de desarrollo, hacer el código fuente más intangible, menos propensa a errores.
- Herramientas de testing integradas; Permite realizar tests unitarios, tests de estrés de aplicaciones ASP.NET y tests de coverage.

- Nueva herramienta de control de código fuente. Sustituye a Sourcesafe y proporciona una experiencia más orientada a la colaboración, con herramientas de merge manual y automático para soportar mejor operaciones de multiple check-outs y branching.
- Just my code debugging. El depurador se puede configurar para que ignore código fuente de 3rd parties y depure sólo nuestro código fuente.
- Smart tags: La funcionalidad más importante de controles (wincontrols y webcontrols) se agrupa en smart tags.
- Edit-and-continue: Permite depurar, cambiar código mientras estamos depurando (en un breakpoint) y en la mayoría de casos, seguir la depuración.
- Auto using: El entorno es capaz “de auto incluir” aquellas sentencias “using” que no se hayan puesto en el código fuente y que sean necesarias para las clases que se estén utilizando.
- Soporte para SQL Server 2005: posibilidad de desarrollar componente para SQL Server 2005, directamente.

### **3.1.5.2 Novedades Principales en los Lenguajes**

- Soporte para genéricos. Los genéricos permiten definir clases parametrizables con parámetros que son de otras clases. Ésta característica está soportada solo a nivel del CLR.
- Partial types. Permiten una misma clase en más de un fichero de código fuente.
- Enumeradores: Facilitan la creación de clases que soporten for... each
- Nullable Types: Implementados gracias a los genéricos, permite que los tipos por valor (Ej. Int32 o Int64) puedan tener el valor null.
- Mejoras especialmente pensadas para p/invoke como la posibilidad de declarar arrays de tamaño fijo.

### **3.1.5.3 Novedades Principales de ASP.NET**

- Master Pages: Permite mantener un layout consistente en un conjunto de páginas de un site.

- Expression syntax: Permite definir expresiones que se evaluarán antes de procesar la página. Se utiliza en el nuevo sistema de localización implementado en ASP.NET y se puede usar también para establecer determinados valores de control de forma dinámica, antes de que la página ASP.NET sea procesada por el Framework.
- Nuevos controles de alto nivel: Controles de login/logout, sitemaps, wizards.
- Herramienta de configuración: Una nueva herramienta de configuración (web) permite configurar el fichero web config sin necesidad de editar el XML.
- Localización basada en ficheros de recursos: Similar a los winforms de Visual Studio .NET 2003
- API para personalización y para webparts.
- Nuevo servidor web integrado. No es necesario tener un IIS instalado en la máquina para crear y depurar aplicaciones ASP.NET.
- Nuevo API de configuración más flexible.

#### **3.1.5.4 Novedades Principales de ADO.NET**

- Permiten abstraer el resto de las aplicaciones del data provider utilizado.
- Objeto DataTableReader que permite iterar para los valores de un DataTable de forma rápida y eficiente.
- Posibilidad de cambiar manualmente el RowState de una o varias filas de un DataTable.

#### **3.1.6 MICROSOFT VISUAL STUDIO 2005 APPLICATION PLATFORM**

La plataforma de Aplicaciones de Microsoft es un portafolio de soluciones, capacidades y mejores prácticas enfocadas en ayudar a las áreas de tecnología a alinearse con su negocio y de esta forma maximizar oportunidades.

##### **3.1.6.1 Common Language Runtime (CLR).**

Punto NET Framework proporciona un entorno en tiempo de ejecución denominado Common Language Runtime, que ejecuta el código y proporciona servicios que facilitan el proceso de desarrollo, el diseño de los componentes y de las

aplicaciones cuyos objetos interactúan entre lenguajes distintos. Los objetos escritos en lenguajes diferentes pueden comunicarse entre sí, lo que permite integrar sus comportamientos de forma precisa. Por ejemplo, puede definir una clase y, a continuación, utilizar un lenguaje diferente para derivar una clase de la clase original o llamar a un método de la clase original. También se puede pasar al método de una clase una instancia de una clase escrita en un lenguaje diferente. Esta integración entre lenguajes diferentes es posible porque los compiladores y las herramientas de lenguajes orientados al motor en tiempo de ejecución utilizan un sistema de tipos común definido por el motor en tiempo de ejecución, y los lenguajes siguen las reglas en tiempo de ejecución para definir nuevos tipos, así como para crear, utilizar, almacenar y enlazar tipos.

#### **3.1.6.2 Acceso a Datos.**

Punto NET Compact Framework ofrece un conjunto robusto de herramientas relacionadas con los datos. La compatibilidad con la base de datos local la proporciona SQL Server CE. En el servidor, .NET Compact Framework ofrece compatibilidad con SQL Server. También ofrece un componente de .NET Compact Framework que permite trabajar con bases de datos de Pocket Access.

#### **3.1.6.3 Aplicaciones Windows Forms.**

Los formularios Windows Forms sirven para crear aplicaciones de Microsoft Windows en .NET Framework. Este marco de trabajo proporciona un conjunto de clases claro, orientado a objetos y ampliable, que permite desarrollar complejas aplicaciones para Windows. Además, los formularios Windows Forms pueden actuar como interfaz de usuario local en una solución distribuida de varios niveles.

#### **3.1.6.4 Aplicaciones Web Forms.**

Los formularios Web Forms son una tecnología ASP.NET que se utiliza para crear páginas Web programables. Los formularios Web Forms se representan como código HTML y secuencias de comandos compatibles con exploradores, lo que permite ver las páginas en cualquier explorador y plataforma.

## **3.2 REPORTING SERVICES 2005**

Una herramienta para mostrar informes, fácil de instalar, que utiliza formato estándar (XML) y que se ve en el visor integrado, viene en el Framework 2.0, integrada en SQL Server 2005 en todas las versiones, es Reporting Services la misma que Microsoft ha facilitado para el tratamiento de Informes en SQL Server 2005 y en aplicaciones .NET 2.0. <sup>5</sup>

### **3.2.1 INTRODUCCIÓN**

Con la orientación de Microsoft hacia una plataforma BI (Business Intelligence) se ha abierto las puertas para nuevas herramientas empresariales tal como administradores de servidores y otros servicios. Con Microsoft SQL Server 2000 o 2005 se incluye un componente adicional el cual contiene un generador de reportes conocido actualmente como Microsoft SQL Server Reporting Services, este componente permite generar reportes de datos almacenados en una base de datos sin necesitar herramientas de terceros. Esta es una herramienta que esta basada en una plataforma de servidor de reportes, la cual es administrada vía Web Services, permitiendo un acceso fácil, seguro y una gran variedad de formatos. Aunque esta herramienta estaba planeada para ser liberada con SQL Server 2005, Microsoft decidió liberar este componente para trabajar con SQL Server 2000. Encontrándose en el mercado en ambas presentaciones.

### **3.2.2 DEFINICIÓN**

Reporting Services es una plataforma de reportes basada en servidores, la misma que puede ser empleada para crear y administrar reportes tabulares, de matrices, gráficos y de libre formato, la información de estos reportes pueden provenir de

---

<sup>5</sup> [http://www.sqlmax.com/reportin\\_services1.asp](http://www.sqlmax.com/reportin_services1.asp)

diferentes orígenes de datos. Los reportes que se definen pueden ser administrados a través de una conexión basada en Web.

### **3.2.3 ARQUITECTURA DE REPORTING SERVICES**

Reporting Services está dividido en:

- Servidor de informes. Es el componente principal y se implementa como un servicio de Microsoft Windows y como un servicio Web. Se encarga de generar los reportes a través de los servicios Web y de la seguridad de los mismos.
- Administrador de informes. Es una herramienta de administración de los informes. Se accede a esta a través del explorador Web (IE), que lleva por debajo un portal Share Point o desde el MSSS Management Studio. A través de este podemos asignar permisos, crear carpetas, ver informes, crear nuevos informes con Report Builder, crear suscripciones(o instantáneas) etc.
- Base de datos del servidor de informes. Es una base de datos SQL Server 2005 donde almacena toda la información que tiene que ver con los informes, con la seguridad, suscripciones, instantáneas y demás extensiones del mismo.
- Herramienta de configuración de Reporting Services. Esta es la encargada de configurar el servidor de informes, entre ellos, el estado del servidor, Directorio virtual del servidor de informes y administrador de informes, Identidad del servicio de Windows, identidad del servicio Web, Instalación de base de datos, claves de cifrado, Inicialización, Configuración de correo electrónico y cuenta de ejecución.



### Arquitectura de Reporting Services

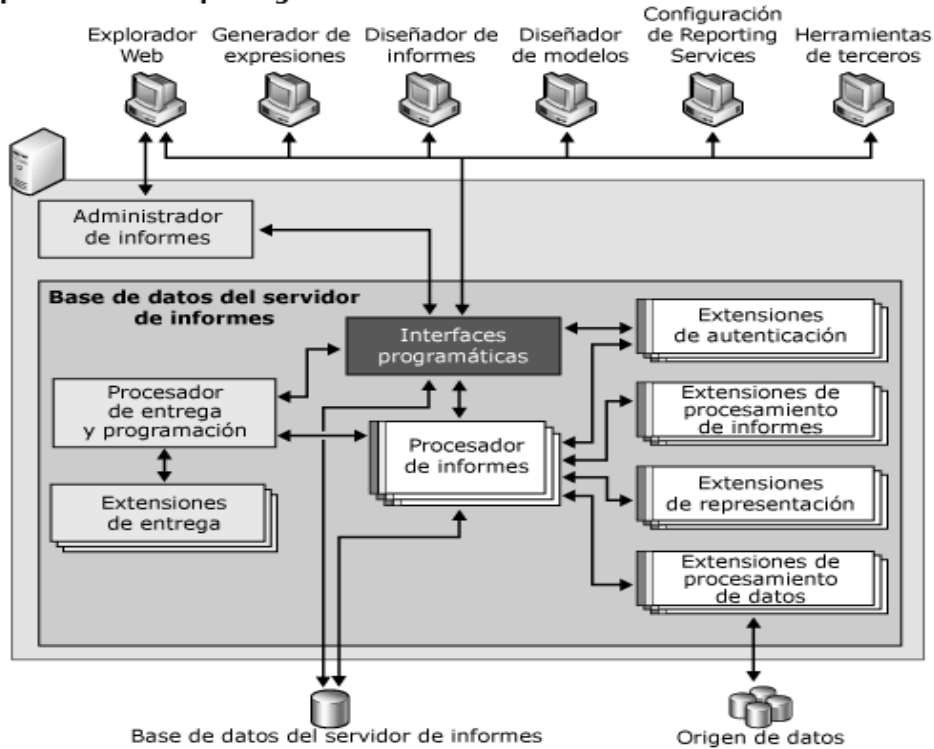


Figura 3.1 Arquitectura de Reporting Services

### 3.2.4 INSTALAR SQL SERVER REPORTING SERVICES

Para instalar Reporting Services se debe cumplir con los siguientes requerimientos:

Hardware

- Cualquier procesador Pentium II Intel o compatible de 500 MHz.
- 256 Mb de RAM, preferible 512 Mb.
- 50 Mb de espacio de disco libre.

Para instalar Reporting Services, se deben seguir los siguientes pasos:

1. Ingresar al sistema con credenciales de Administrador, luego insertar el disco de instalación de Reporting Services y hacer doble clic en Setup.exe.
2. Aceptar las condiciones de de licenciamiento y presionar Next.
3. Aparece un diálogo que muestra el estado de recopilación de información necesaria para instalar Reporting Services, presionar Next.

4. Aparece un diálogo donde se muestra un mensaje que confirma la existencia de los pre-requisitos, presionar Next.
5. Aparece un diálogo donde se muestra un mensaje que confirma la existencia de los pre-requisitos, presionar Next. Aparece el mensaje del Windows Installer y a continuación un dialogo, presionar Next.
6. Ingresar la información de registro, luego pulsar Next.
7. Seleccionar los componentes a instalar, luego presionar Next.
8. Aparece un diálogo donde se elegirán las credenciales para la instalación (crear una cuenta de Dominio e incluirla en el grupo Domain Administrators).
9. Pulsar Next, aparece un diálogo que muestra los directorios virtuales a crearse, presionar Next
10. Observar un diálogo que permite seleccionar la instancia de SQL Server, que se empleará con Reporting Services, presionar Next.(SQL creará una BD para el ReportServer y ReportServer TempDB)
11. Ingresar la información necesaria para enviar los reportes a través de email.
12. Especificar la instancia SQL, donde se creará la BD de ejemplo (AdventureWorks), presionar Next.
13. Seleccionar el modo de licenciamiento, presionar Next.
14. Presionar el botón Install.
15. Luego de unos instantes que el instalador realiza todas las tareas necesarias aparece el dialogo indicando el fin de la instalación, presionar Finish y la instalación habrá concluido.

### **3.2.5 REPORT DEFINITION LANGUAGE RDL**

El RDL es el Report Definition Language, es generado en XML, y brinda muchas posibilidades a la hora de trabajar con él, ya que permite programar a mano desde código o pasarlo a través de firewalls al ser texto plano.

#### **3.2.5.1 Etiquetas**

El RDL está constituido por una serie de etiquetas clasificadas en:

- Cabecera. En la cabecera podemos encontrar la versión de XML, la codificación.
- Report. Contiene la dirección de la definición del RDL y todo el contenido del informe va entre sus etiquetas.
- DataSources. Muestra la fuente de datos, el proveedor de datos, la cadena de conexión entre otros.
- ReportParameters. Se define el nombre del parámetro, el tipo y el texto que mostrará al usuario.
- PageHeader. Cabecera del Informe.
- Body. Este es el cuerpo del Informe, aquí es donde van los elementos del informe, como las especificaciones de diseño y los ReportItems que contiene las tablas, textbox y demás ítems del informe.
- PageFooter. Esta es el pie del informe.
- Propiedades de Página. Ancho, alto, márgenes, idioma y todo lo relacionado con el Report en general.
- DataSets. Contiene los DataSet/s utilizados para el informe. En estos se pueden encontrar: los campos y sus tipos y la consulta empleada entre otros.

### **3.2.6 DISEÑAR Y CREAR INFORMES**

Para crear un informe con Reporting Services existen dos formas:

- Con Visual Studio .NET 2005, añadiendo un nuevo proyecto de servidor de informes. A este se le agrega una fuente de datos o DataSet y un informe nuevo donde se va a diseñar el reporte.
- Con Report Builder, Deben estar creadas las fuentes de datos o DataSets ya que debe ser muy fácil de entender porque esta orientado a usuarios finales.

### **3.2.7 ADMINISTRAR REPORTING SERVICES**

Al instalar Reporting Services se instalan varios componentes de servidor que se deben administrar y mantener como: el servicio Web del servidor de informes, el

servicio de Windows del servidor de informes y la base de datos del servidor de informes.

Para administrar por completo una instalación de Reporting Services, debe disponer de los siguientes permisos:

- Ser miembro del grupo local de administradores en el equipo del servidor de informes.
- Tener permisos de administrador de base de datos para la instancia de SQL Server que aloja la base de datos.

Para administrar el servidor de informes existen varias herramientas que se pueden utilizar:

#### Herramientas de SQL Server 2005

- Administrador de configuración de SQL Server, se utiliza para iniciar o detener el servicio Windows del Servidor de informes y para configurar propiedades avanzadas del servicio.
- Configuración de superficie de SQL Server, sirve para habilitar solicitudes de servicios Web y el Procesador de entrega y programación. Esta herramienta permite controlar los tipos de operaciones que puede efectuar el servidor de informes. Permite desactivar operaciones de servicios Web, programación y entrega para realizar el mantenimiento del sistema, investigar problemas o cambiar la configuración de una implementación.
- SQL Server Management Studio se utiliza para administrar instancias del servidor de informes en un entorno único.

#### Herramientas de Reporting Services

- Administrador de informes, sirve para administrar un servidor de informes en una conexión basada en Web. El Administrador de informes se abre en un

explorador, por lo que resulta práctico para ejecutarlo desde equipos remotos.

- Configuración de Reporting Services, permite implementar un servidor de informes después de su instalación o para volver a configurar las opciones definidas durante la instalación.
- Utilidades de línea de comandos de Reporting Services, incluye tres utilidades de línea de comandos: un host de secuencias de comandos (rs.exe), una herramienta de configuración (rsconfig.exe) y una herramienta de administración de claves (rskeymgmt.exe). Se puede utilizar la herramienta de configuración de Reporting Services en lugar de rsconfig y rskeymgmt. La utilidad rs sirve para ejecutar secuencias de comandos en un servidor de informes.

#### Herramientas de Microsoft Windows

- Administrador de servicios Internet, sirve para modificar la seguridad en los directorios virtuales utilizados para obtener acceso al servidor de informes y al Administrador de informes.
- Visor de sucesos y consola Rendimiento permite supervisar el rendimiento del servidor de informes y el Administrador de informes.

## **IV. CAPITULO**

### **DESARROLLO DEL SISTEMA REGASIS**

#### **4.1 CONCEPTUALIZACIÓN**

##### **4.1.1 NEGOCIACIÓN**

###### **4.1.1.1 ENTREVISTAS**

#### **Función de la Plantilla**

La función de esta plantilla es estandarizar y formalizar todos los aspectos dentro de una entrevista inicial hacia un cliente (la reunión inicial para la posterior creación de la propuesta de trabajo) para lograr un documento formal de inicio del proyecto.

#### **Objetivos de la Plantilla**

1. Estandarizar la entrevista hacia el cliente.
2. Crear una estructura para las entrevistas iniciales.

#### **Alcance de la Plantilla**

Esta plantilla sólo se usará en la primera reunión con el cliente y responderá las preguntas hechas en la declaración de trabajo (SOW) y algunas de la especificación de requerimientos (ERS).

Las preguntas descritas aquí son preguntas estándar.

## Revisiones de Plantilla

<b>Versión</b>	<b>Fecha de Entrega</b>	<b>Revisado por</b>
1.0.0	2007-11-07	Esteban Oviedo

## Control de versiones Plantilla

<b>Versión</b>	<b>Encargado</b>	<b>Descripción</b>
1.0.0	Elías Conejo Cantillo	Creación de plantilla

## Carta de Aprobación

Por medio de la firma de esta carta de aceptación se aprueba la información de esta entrevista.

---

**Esteban Oviedo Blanco**

Director de Proyectos / Chief Projects Officer

## **Entrevista**

El objetivo de esta entrevista es crear un alcance y una definición del proyecto a realizar.

### **Persona a entrevistar**

Ing. Esteban Oviedo Blanco, Director de Proyectos, Babel Software.

### **Fecha de la entrevista**

Lunes 12, de abril de 2008

### **Lugar**

Sala de Juntas de Babel Software



## Justificación

1. ¿Por qué se pretende realizar este proyecto?

Porque este proyecto constituirá una forma de agilizar el proceso de registro de horas de entrada y salida de los empleados de babel, por lo que facilitará además el ingreso de ausencias y su justificación, controles que en este momento no se manejan.

2. ¿Qué se pretende con la realización de este proyecto?

Lo que se pretende es sencillamente llevar un control más detallado sobre la distribución de las horas de los empleados en diferentes momentos del día.

3. El producto realizado en este proyecto, ¿en qué beneficiará a su empresa?

Este proyecto forma parte del siguiente objetivo estratégico:

- Crear e implementar herramientas que permitan la ejecución de técnicas para incrementar el control y el seguimiento de los resultados en todos los proyectos internos, así como en los proyectos para con nuestros clientes. (Plan estratégico 2008 - 2009)

Es nuevo sistema a desarrollos contribuirá al mejoramiento de los procesos internos de la empresa.

4. ¿Qué problemas presenta la empresa a falta del proyecto que se pretende realizar?

En un inicio los registros de asistencia (entrada y salida) se realizaban manualmente, es decir, el funcionario de babel ingresaba su nombre e indicaba la hora de ingreso y salida. Pero esta forma de hacer las cosas tiene muchos inconvenientes.

Luego se desarrolló un pequeño sistema de control de asistencia, pero este sufre de fallas constantes y esta centralizado. Además de que tiene muchas limitaciones.

5. ¿Qué área (s) de la empresa se vería(n) beneficiada(s) con el proyecto?

La parte beneficiada será el departamento de recursos humanos ya que se le facilitará el trabajo de evaluación de desempeño y puntualidad de cada empleado.

6. ¿Qué posibles problemas podrían presentarse en la empresa como consecuencia del proyecto?

La resistencia al cambio, que conlleva a un control mucho más estricto del que se venía haciendo anteriormente.

## **Requerimientos Funcionales**

1. ¿Qué funciones se desea que el sistema realice?

El sistema debe permitir que el funcionario de Babel lleve su propio control de asistencia y ausencias. Las cuales deberán estar justificadas.

Debe ser muy intuitivo y de interfaz amigable.

2. ¿De qué otros sistemas dependería el sistema a desarrollar?

Este sistema como tal es independiente, pero se debe conectar a una base de datos centralizada llamada "BabelSys" que es la usada por todos los sistemas internos de la empresa.

Si es posible, el sistema deberá reutilizar la lógica desarrollada en "BabelSys": procedimientos almacenados, funciones, etc.

3. ¿Sobre qué procesos influirá la ejecución del sistema?

Se modificará el proceso actual de control de asistencia descentralizándose y permitiendo de esta forma que cada funcionario se encargue de su propio control.

## **Requerimientos de Calidad**

1. ¿Qué características se desean en el producto a realizar?

Sistema debe ser muy robusto y muy estable, ya que las condiciones en cada computadora difieren, éste debe ejecutarse sin problema alguno.

Por otro lado los sistemas internos de Babel constantemente están siendo modificados o actualizados en cuanto a funcionalidad, así que esos cambios de plataformas o de bases de datos no deben afectar a la ejecución normal de este nuevo sistema.

## Requerimientos de Costo

1. ¿Ustedes disponen de un presupuesto para este producto?

Si. Esta inversión ya fue contemplada por la junta directiva y ya fue justificada a la misma. Ya se cuenta con un presupuesto base.

2. ¿En caso de que el proyecto sufra un cambio de forma que necesite implementarse alguna otra funcionalidad o añadir cosas que hayan podido quedar por fuera y se necesite una ampliación, la empresa cuenta con el presupuesto para correr con el gasto?

Si. Como le mencioné este proyecto forma parte de un plan estratégico y es imprescindible para nosotros el materializarlo cuanto antes.

## Requerimientos de Seguimiento

1. ¿Qué tipo de entregable quiere que les entreguemos?

- a. Informes de seguimiento por correo electrónico.
- b. Documento de análisis y diseño.
- c. Prototipo.
- d. Producto funcional.

2. ¿Cada cuánto desea que se realicen reuniones para informar sobre el estado de avance del proyecto?

De forma semanal cada viernes y por medio del correo electrónico deseo recibir noticias sobre el transcurso del desarrollo del proyecto.

## Requerimientos Tecnológicos

1. ¿Se desea alguna plataforma de desarrollo en especial?

Si. El sistema debe estar desarrollado en .Net 2005 y deberá ejecutarse sobre Windows XP y Vista.

El sistema, además deberá acceder a una base de datos por medio de Servicios Web. Deberá contener además una parte web para reportes el cual deberá poder visualizarse adecuadamente en Internet Explorer 7 o superior.

2. ¿Se desea algún sistema administrador de base de datos en especial?

Si. El sistema de base de datos que se usará es SQL Server 2005.

3. ¿Se desea algún lenguaje de programación en especial?

Si. El lenguaje de programación a usar será C# en .Net 2005. Se usarán Win y Web forms, Servicios Web y Reporting Services.

4. ¿Se desea utilizar software libre o se poseen licencias para el software que se desea utilizar?

El sistema será desarrollado en su totalidad con tecnologías de Microsoft y si se cuenta con las respectivas licencias.

5. ¿Se desea alguna arquitectura de desarrollo en especial?

Si. Quiero una arquitectura multicapa orientada al SOA (Arquitectura Orientada a Servicios).

6. ¿Se tiene algún estándar de programación, base de datos u otros que se quiera que Babel Software siga?

Si. Esos estándares se encuentran debidamente documentados y están a disposición de todos los funcionarios de la empresa. Esos son los que se deberán seguir.

## **Restricciones**

1. ¿Existen límites dentro de lo que se espera del producto a realizar?

Babel software tiene oficinas en diferentes partes (Ecuador, Estados Unidos y varias en Costa Rica). Este sistema solo está dirigido para las oficinas centrales en Costa Rica.

2. ¿Existen limitaciones sobre donde debe desarrollarse el software (instalaciones cliente, instalaciones de Babel)?

El sistema se desarrollará aquí en las oficinas centrales de Babel.

3. ¿Se está anuente a realizar una inversión en cuanto a infraestructura de servidores se refiere, para poder brindarle un ambiente ideal a la aplicación?

Si. Si hubiera alguna inversión extra Babel correría con los gastos.

#### 4.1.1.2 DECLARACIÓN DE TRABAJO



## Declaración de Trabajo (SOW)

---

**REGASIS**

**Versión 1.0.0**

**Diana Carolina Reyes Reyes**

**Fecha (2008-04-14)**

## **Función de la Plantilla**

La función de esta plantilla es estandarizar y formalizar todos los aspectos dentro de una declaración de trabajo para lograr un documento formal de inicio del proyecto. Para esto se utilizarán estándares mundiales.

## **Objetivos de la Plantilla**

1. Estandarizar la declaración de trabajo.
2. Crear una estructura para la declaración de trabajo.

## **Alcance de la Plantilla**

Esta plantilla sólo se usará después de la primera reunión con el cliente y se obtuvieron una serie de requerimientos iniciales, cuando se tenga que especificar qué deberá hacer el sistema a crear y bajo qué ambiente estará (políticas, etc.).

## Revisiones de Plantilla

<b>Versión</b>	<b>Fecha de Entrega</b>	<b>Revisado por</b>
1.0.0	2007-11-07	Esteban Oviedo

## Control de versiones Plantilla

<b>Versión</b>	<b>Encargado</b>	<b>Descripción</b>
1.0.0	Elías Conejo Cantillo	Creación de plantilla

## Revisiones

<b>Versión</b>	<b>Fecha de Entrega</b>	<b>Revisado por</b>
1.0.0	2008-05-01	Erick Salazar

## Control de versiones

<b>Versión</b>	<b>Encargado</b>	<b>Descripción</b>
1.0.0	Diana Carolina Reyes Reyes	Creación de documento

## Carta de Aprobación

Por medio de la firma de esta carta de aceptación se aprueba la información de esta declaración de trabajo.

---

**Ing. Esteban Oviedo Blanco**

Director de Proyectos / Chief Projects Officer

## Tabla de Contenido

<b>Función de la Plantilla .....</b>	<b>- 58 -</b>
<b>Objetivos de la Plantilla.....</b>	<b>- 58 -</b>
<b>Alcance de la Plantilla.....</b>	<b>- 58 -</b>
<b>Revisiones de Plantilla .....</b>	<b>- 59 -</b>
<b>Control de versiones Plantilla.....</b>	<b>- 59 -</b>
<b>Revisiones .....</b>	<b>- 59 -</b>
<b>Control de versiones.....</b>	<b>- 59 -</b>
<b>Carta de Aprobación .....</b>	<b>- 59 -</b>
<b>Tabla de contenido.....</b>	<b>- 60 -</b>
<b>REGASIS .....</b>	<b>- 61 -</b>
Producto .....	- 61 -
Justificación .....	- 61 -
Resumen de las especificaciones del producto.....	- 61 -
<b>Requerimientos Funcionales y de Desempeño.....</b>	<b>- 61 -</b>
<b>Requerimientos de Calidad.....</b>	<b>- 62 -</b>
<b>Requerimientos de Costo.....</b>	<b>- 62 -</b>
<b>Requerimientos de Seguimiento .....</b>	<b>- 62 -</b>
<b>Requerimientos Tecnológicos.....</b>	<b>- 62 -</b>
<b>Restricciones.....</b>	<b>- 63 -</b>
Beneficios .....	- 64 -
<b>Beneficios generales .....</b>	<b>- 64 -</b>
<b>Beneficios para Babel.....</b>	<b>- 64 -</b>
<b>Beneficios para los usuarios de Babel .....</b>	<b>- 64 -</b>



## **REGASIS**

Sistema para el registro del control de entradas, salidas y ausencias para los funcionarios de Babel Software.

### **Producto**

Realizar un proceso de registro ágil y rápido y un mejor control de entradas, salidas y ausencias de los funcionarios de Babel software.

### **Justificación**

1. Mejorar los procesos de gestión de recursos humanos.
2. Descentralizar el proceso de registro de entradas y salidas.
3. Implementar un control detallado de ausencias.
4. Crear e implementar herramientas que permitan la ejecución de técnicas para incrementar el control y el seguimiento de los resultados en todos los proyectos internos, así como en los proyectos para con nuestros clientes.  
(Plan estratégico 2008 - 2009)
5. Contar con información rápida y ágilmente para conocer la situación de ingresos-salidas de cada funcionario.

### **Resumen de las especificaciones del producto**

#### **Requerimientos Funcionales y de Desempeño**

1. Se debe de crear un patrón de proceso genérico para el registro de entradas y salidas.
2. Se debe descentralizar la forma de ingreso de datos.
3. Se debe crear un proceso para llevar el control de ausencias y su respectiva justificación.
4. Se debe permitir el acceso desde cualquier computadora de la empresa que esté registrada en el dominio y cuyo usuario posea cuenta en Active Directory.
5. Se deben permitir operaciones simultáneamente sobre la base de datos ya que los funcionarios podrán ejecutar operaciones con el sistema al mismo tiempo.
6. Se debe tener la disponibilidad del sistema en cualquier momento del día.

7. El sistema debe ejecutarse siempre desde la computadora del usuario en forma ininterrumpida y no debe interferir en otros procesos del sistema operativo u otros que interfieran con su rendimiento.
8. Se debe instalar en clientes con Windows XP y Vista.
9. Se debe implementar usando el .Net Framework 2.0
10. El software debe ser modular y de fácil mantenimiento.

### **Requerimientos de Calidad**

1. Se debe proyectar un sistema amigable al usuario y no propenso a fallos de conexión ni de software.
2. Se debe poder expandir su funcionamiento y su arquitectura no debe variar.
3. Se debe esperar un tiempo de respuesta corto ante cada ejecución de procesos.

### **Requerimientos de Costo**

1. Habrá que apegarse al presupuesto para la realización del proyecto, el cual consiste en 6 meses.
2. Si hubiera necesidad de realizar inversión extra (equipos, redes, servidores, etc.) estos tendrán que cumplir con los estándares mínimos.
3. El sistema no debe superar 6 meses de tiempo en su desarrollo.

### **Requerimientos de Seguimiento**

1. Se deben de presentar informes parciales de avance, en términos de avance del proyecto.
2. Se deben dar reuniones de revisión cuando el administrador de proyectos así lo solicite.
3. Todo cambio al sistema y al proceso de seguimiento debe ser justificado.
4. Se debe mostrar el avance con revisiones periódicas del software.

### **Requerimientos Tecnológicos**

1. Se debe de implantar en computadoras con Windows XP y Vista.
2. Se debe desarrollar usando .Net 2005, C#, Web Services, SQL Server 2005, Reporting Services, Win y Web Forms (ASPX) y Ajax.

3. Se debe hacer uso de la red interna de Babel para la recepción y transmisión de datos.

## **Restricciones**

1. Las computadoras deberán de tener al menos las siguientes características:
  - a. Procesador Pentium a 233 megahercios (MHz) o mayor velocidad (se recomienda 300 MHz)
  - b. Al menos 64 megabytes (MB) de RAM (se recomienda 128 MB)
  - c. Un mínimo de 1,5 gigabytes (GB) de espacio disponible en el disco duro
  - d. Unidad de CD-ROM o DVD-ROM
  - e. Un teclado y un mouse de Microsoft, o algún otro dispositivo señalador compatible
  - f. Adaptador de vídeo y monitor con una resolución Super VGA (800 x 600) o mayor.
  - g. Adaptador de red.
2. Se debe esperar un tiempo de respuesta no mayor a 5 segundos.
3. Los servidores Web y de base de datos deberán tener las siguientes características como mínimo:
  - a. Al menos un procesador de 133 MHz; se recomienda un procesador de 550 MHz o más veloz; soporte para hasta cuatro procesadores en un mismo servidor.
  - b. Memoria: Por lo menos 128 MB de RAM; se recomienda memoria de 256 MB o más hasta un máximo de 4 GB.
  - c. Disco rígido: Entre 1.25 y 2 GB de espacio disponible en el disco rígido.
  - d. Lector: lector de CD-ROM o DVD-ROM.
  - e. Monitor VGA o hardware que admita la redirección de consola; se recomienda un Super VGA con resolución 800 x 600 o superior.
4. Se debe instalar en cada computadora o servidor el .Net Framework 2.0.
5. Cada computadora debe tener configurada el acceso a la red interna de Babel.

6. Los usuarios deben tener una cuenta en Active Directory, así como permisos de escritura sobre algunas tablas de la base de datos.

## **Beneficios**

### **Beneficios generales**

1. Mejor control de gestión de recursos humanos.
2. Margen de error en verificación 0 %.

### **Beneficios para Babel**

1. Tecnología de punta en identificación de personal.
2. Mayor productividad debido a la eliminación de procesos manuales.
3. Respuesta inmediata en consultas online de tiempos y asistencias.
4. Reducción de costos en la renovación de tarjetas magnéticas u otros dispositivos de marcación.
5. Soporta el crecimiento de su personal en múltiples áreas, sin disminuir el rendimiento del sistema.
6. Ahorro en infraestructura de implementación, ya que el equipo funciona con tecnología TCP/IP, lo que garantiza la escalabilidad del sistema.
7. Ahorro de pago por horas normales o extras no reales.

### **Beneficios para los usuarios de Babel**

1. Rapidez en el proceso de información.
2. Ahorro de tiempo por la eliminación de tareas manuales.
3. Obtención de reportes y consultas online, rápidos y eficientes.
4. Reportes Gerenciales para toma de decisiones y medición de producción.
5. Información 100% segura de las marcaciones, sin opciones a fraudes.
6. Oportuna intervención en la toma de decisiones.

## 4.1.2 ESPECIFICACIÓN DE REQUERIMIENTOS



# Especificación de Requerimientos

---

**REGASIS**

**Versión 1.0.0**

**Diana Carolina Reyes Reyes**

**2008-05-28**

## **Función de la Plantilla**

La función de esta plantilla es estandarizar y formalizar todos los aspectos dentro de una especificación de requerimientos, ejemplo: casos de uso. Para esto se utilizarán estándares mundiales (IEEE, UML).

## **Objetivos de la Plantilla**

1. Estandarizar la especificación de requerimientos en base al estándar IEEE 830
2. Especificar los requerimientos funcionales (funciones) con el estándar UML
3. Crear una estructura para la especificación de requerimientos
4. Ayudar en el rastreo de requerimientos

## **Alcance de la Plantilla**

Esta plantilla sólo se usará después de la toma de requerimientos, cuando se tenga que especificar qué deberá hacer el sistema a crear y bajo qué ambiente estará (políticas, etc.)

## Revisiones de Plantilla

<b>Versión</b>	<b>Fecha de Entrega</b>	<b>Revisado por</b>
1.0.0	2008-05-28	Michael Bogantes Fabio Fernández
1.0.0	2008-05-28	Esteban Oviedo

## Control de versiones Plantilla

<b>Versión</b>	<b>Encargado</b>	<b>Descripción</b>
1.0.0	Elías Conejo Cantillo	Creación de plantilla

## Revisiones

<b>Versión</b>	<b>Fecha de Entrega</b>	<b>Revisado por</b>
1.0.0	2007-09-13	Erick Salazar

## Control de versiones

<b>Versión</b>	<b>Encargado</b>	<b>Descripción</b>
1.0.0	Diana Carolina Reyes Reyes	Creación de documento

## Carta de Aprobación

Por medio de la firma de esta carta de aceptación los requerimientos aquí descritos se aprueban y serán desarrollados. Se acepta que no quedan requerimientos fuera del alcance del proyecto.

---

**Ing. Esteban Oviedo Blanco**

Director de Proyectos / Chief Projects Officer

## Tabla de contenido

<b>Función de la Plantilla</b> .....	- 66 -
<b>Objetivos de la Plantilla</b> .....	- 66 -
<b>Alcance de la Plantilla</b> .....	- 66 -
<b>Revisiones de Plantilla</b> .....	- 67 -
<b>Control de versiones Plantilla</b> .....	- 67 -
<b>Revisiones</b> .....	- 67 -
<b>Control de versiones</b> .....	- 67 -
<b>Carta de Aprobación</b> .....	- 67 -
<b>Tabla de contenido</b> .....	- 68 -
<b>Introducción</b> .....	- 69 -
Propósito .....	- 69 -
Ámbito del Sistema .....	- 69 -
<b>Beneficios</b> .....	- 69 -
<b>Objetivos</b> .....	- 70 -
<b>Actores</b> - 71 -	
Definiciones, Acrónimos y Abreviaturas .....	- 71 -
Visión General del Documento .....	- 71 -
<b>Descripción General</b> .....	- 72 -
Perspectiva del Producto .....	- 72 -
Funciones del Producto .....	- 72 -
Características de los Usuarios Finales .....	- 73 -
Restricciones .....	- 73 -
Suposiciones y Dependencias .....	- 74 -
Requerimientos Futuros .....	- 75 -
<b>Requerimientos Específicos</b> .....	- 75 -
Interfaces Externas .....	- 75 -
Requerimientos funcionales .....	- 76 -
Requerimientos No Funcionales .....	- 77 -
Requerimiento de Rendimiento .....	- 78 -
Diagramas General de Casos de Uso .....	- 79 -
Casos de Uso de Alto Nivel .....	- 80 -



## **Introducción**

Los problemas que ha sufrido Babel Software por no contar con un proceso claro y automatizado para el control de horarios de empleados han sido constantes y muy variables.

Anteriormente se trató en diversas formas de solucionar este problema, pero sin resultados aceptables. Es por eso que se decidió emprender el proceso de desarrollo de una herramienta capaz de gestionar eficazmente las horas de la jornada laboral del personal.

El presente documento describe en forma general, claro y conciso, los requerimientos para el sistema REGASIS a desarrollar para Babel Software.

## **Propósito**

Servir de guía para la especificación general de los requerimientos funcionales y no funcionales, de tal forma que sea una guía para la comprensión de los procedimientos que se desarrollarán y servirán de base para las fases de conceptualización posteriores.

Este documento va dirigido a los funcionarios de Babel que participan activamente en el proyecto de desarrollo y etapas de análisis y diseño.

## **Ámbito del Sistema**

El sistema REGASIS abarcará el proceso de registro de horas de entrada y salida para la jornada laboral establecida en las oficinas centrales de Babel Software. Así mismo, contempla las ausencias y las justificaciones de las mismas.

## **Beneficios**

### **Beneficios generales**

1. Mejor control de gestión de recursos humanos.
2. Margen de error en verificación 0 %.

### **Beneficios para Babel**

1. Tecnología de punta en identificación de personal.
2. Mayor productividad debido a la eliminación de procesos manuales.
3. Respuesta inmediata en consultas online de tiempos y asistencias.
4. Reducción de costos en la renovación de tarjetas magnéticas u otros dispositivos de marcación.
5. Soporta el crecimiento de su personal en múltiples áreas, sin disminuir el rendimiento del sistema.
6. Ahorro en infraestructura de implementación, ya que el equipo funciona con tecnología TCP/IP, lo que garantiza la escalabilidad del sistema.
7. Ahorro de pago por horas normales o extras no reales.

### **Beneficios para los usuarios de Babel**

1. Rapidez en el proceso de información.
2. Ahorro de tiempo por la eliminación de tareas manuales.
3. Obtención de reportes y consultas online, rápidos y eficientes.
4. Reportes Gerenciales para toma de decisiones y medición de producción.
5. Información 100% segura de las marcaciones, sin opciones a fraudes.
6. Oportuna intervención en la toma de decisiones.

### **Objetivos**

<b>OBJ-001</b>	<b>Brindar Seguridad</b>
<b>Descripción</b>	El módulo a desarrollar debe permitir el ingreso de personas debidamente autorizadas desde las oficinas centrales de Babel
<b>Importancia</b>	Alta
<b>Comentarios</b>	Es necesario controlar el acceso a los diferentes módulos del sistema y evitar accesos no deseados.

<b>OBJ-002</b>	<b>Gestionar ágilmente el control de horas de los empleados</b>
<b>Descripción</b>	REGASIS debe permitir llevar el control de horas eficientemente.
<b>Importancia</b>	Alta
<b>Comentarios</b>	Es imprescindible llevar un control sobre horas trabajadas, de ocio y ausencias.

### Actores

<b>ACT-001</b>	<b>Funcionario</b>
<b>Descripción</b>	Es un empleado de Babel Software que tiene acceso a una computadora y que esta sujeto a un horario laboral.
<b>Comentarios</b>	Es el encargado de ingresar o consultar las horas de entrada y salidas.

<b>ACT-002</b>	<b>Sistema</b>
<b>Descripción</b>	Es el sistema REGASIS.
<b>Comentarios</b>	Es el encargado de realizar algunas consultas y validaciones.

### Definiciones, Acrónimos y Abreviaturas

1. **ERS:** Especificación de Requerimientos de Software.
2. **REGASIS:** Sistema de Registro de Asistencias.

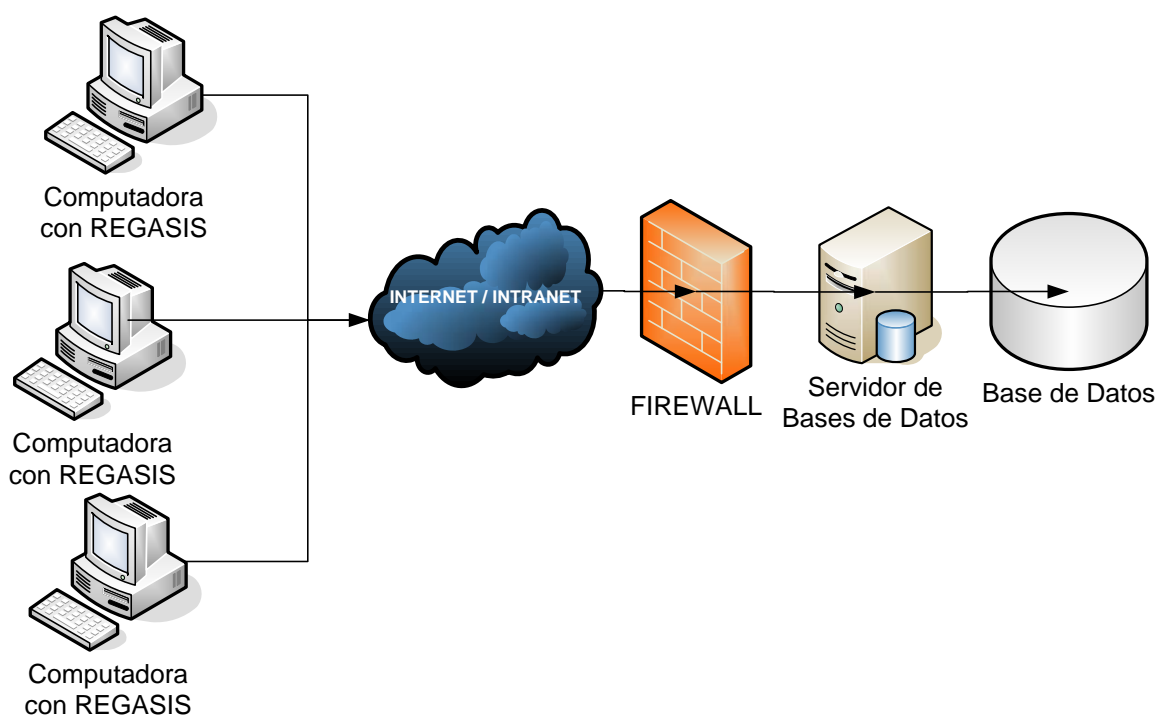
### Visión General del Documento

En las siguientes secciones se describirán las funciones de los sistemas, así como se diagramarán los diferentes casos de uso del mismo.

## Descripción General

### Perspectiva del Producto

REGASIS será un sistema totalmente independiente en el que se instalará a cada cliente. Este tendrá acceso por medio de un Servicio Web publicado en Internet o Intranet, el cual será el encargado de realizar la conexión con el servidor de base de datos.



**Figura 4.1 REGASIS**

Cada usuario deberá poseer una cuenta en Active Directory y los permisos de acceso respectivos.

### Funciones del Producto

Este sistema de registro de control, soportará el mantenimiento de horas de entrada y salida, mostrará el estado del funcionario de babel (si ya marcó la hora de entrada o salida) y facilitará el recordatorio de realizar el proceso anterior ya que será el empleado, el único perjudicado por no hacerlo.

## Características de los Usuarios Finales

Los usuarios de REGASIS son todos aquellos empleados de Babel Software que laboran en las oficinas centrales y que por la naturaleza de su trabajo, deben poseer una computadora.

Los grupos de usuarios son los siguientes (cada nombre también hace referencia a su departamento):

- Administrativos
- Base de Datos
- Comercial
- Desarrolladores
- Directores
- Soporte técnico
- Administradores de proyectos
- QA
- Recursos humanos
- Soporte de aplicaciones

## Restricciones

<b>RST-001</b>	<b>Uso de la base de datos de Babel</b>
<b>Descripción</b>	El sistema se desarrollará usando la base de datos de Babel.
<b>Importancia</b>	Alta
<b>Comentarios</b>	NA

<b>RST-002</b>	<b>Lenguaje de programación</b>
<b>Descripción</b>	El sistema se desarrollará en .NET C# 2005
<b>Importancia</b>	Alta
<b>Comentarios</b>	NA

<b>RST-003</b>	<b>Arquitectura Orientada a Servicios (SOA)</b>
----------------	---

<b>Descripción</b>	Uso obligatorio de servicios tanto de Internet como de Intranet.
<b>Importancia</b>	Alta
<b>Comentarios</b>	NA

<b>RST-004</b>	<b>Uso de la modularidad</b>
<b>Descripción</b>	El sistema debe ser modular, de fácil mantenimiento y con componentes reutilizables.
<b>Importancia</b>	Alta
<b>Comentarios</b>	NA

### Suposiciones y Dependencias

<b>SUP-001</b>	<b>Sistema Operativo</b>
<b>Descripción</b>	El sistema operativo que se utilizará es Windows XP, Windows Vista y Windows 2003 Server
<b>Importancia</b>	Alta
<b>Comentarios</b>	Si se cambia el sistema operativo, provocará nuevos requerimientos y posibles fallas al sistema.

<b>SUP-002</b>	<b>.Net Framework</b>
<b>Descripción</b>	Todas las computadoras deberán contar con el .Net Framework 2.0 instalado.
<b>Importancia</b>	Alta
<b>Comentarios</b>	Si no la tienen el sistema no se ejecutará.

<b>SUP-003</b>	<b>Usuarios registrados y empleados de Babel</b>
<b>Descripción</b>	Todos los empleados a los que se les instalará el sistema son empleados de Babel Software y cuentan con permisos de Active Directory.
<b>Importancia</b>	Alta
<b>Comentarios</b>	NA

## Requerimientos Futuros

Este sistema se podrá integrar con otros módulos ya desarrollados (ejemplo: Portal Interno de Babel) y le será posible la implementación de mejoras y actualizaciones.

REGASIS podrá integrarse con el sistema de contabilidad y de recursos humanos. Adicionalmente se valorará la posibilidad de extenderle varios módulos para llevar el control de todos aquellos funcionarios que no laboren dentro de las oficinas centrales de Babel.

## Requerimientos Específicos

### Interfaces Externas

<b>IEX-001</b>	<b>Base de datos de Babel</b>
<b>Descripción</b>	El sistema a realizar necesita consultar (obtener) los empleados de la base de datos de Babel Software.
<b>Importancia</b>	Alta
<b>Existencia</b>	No existe
<b>Funciones</b>	A crear
<b>A realizar por</b>	Babel Software
<b>Comentarios</b>	Esta interfaz será un “WebService” que obtendrá de la Base de Datos la información del empleado.

<b>IEX-002</b>	<b>Active Directory</b>
<b>Descripción</b>	El sistema a realizar necesita obtener los empleados de Active Directory de Babel Software.
<b>Importancia</b>	Alta
<b>Existencia</b>	No existe
<b>Funciones</b>	A crear
<b>A realizar por</b>	Babel Software

<b>Comentarios</b>	NA.
--------------------	-----

## Requerimientos funcionales

<b>RQF-001</b>	<b>Registrar hora de entrada y salida laboral</b>
<b>Descripción</b>	El sistema debe de identificar el perfil del usuario para mostrarle las horas de entrada y permitirle el registro de las mismas.
<b>Objetivo</b>	OBJ-001
<b>Importancia</b>	Alta
<b>Estado</b>	Aprobado
<b>Estabilidad</b>	Alta
<b>Comentario</b>	Para la obtención del perfil del usuario se requiere interactuar con Active Directory.

<b>RQF-002</b>	<b>Registrar hora de entrada y salida de almuerzos</b>
<b>Descripción</b>	El sistema debe de identificar el perfil del usuario para permitirle el registro de salidas y entradas después de la hora de almuerzo.
<b>Objetivo</b>	OBJ-001
<b>Importancia</b>	Alta
<b>Estado</b>	Aprobado
<b>Estabilidad</b>	Alta
<b>Comentario</b>	Para la obtención del perfil del usuario se requiere interactuar con Active Directory.

<b>RQF-003</b>	<b>Registrar, consultar y justificar ausencias</b>
<b>Descripción</b>	El sistema debe permitir al empleado de Babel registrar, consultar e ingresar ausencias y justificarlas.
<b>Objetivo</b>	OBJ-001
<b>Importancia</b>	Alta
<b>Estado</b>	Aprobado
<b>Estabilidad</b>	Alta
<b>Comentario</b>	Para la obtención del perfil del usuario se requiere interactuar con



	Active Directory.
--	-------------------

<b>RQF-004</b>	<b>Validar o consultar empleados</b>
<b>Descripción</b>	El sistema debe validar los usuarios que usan el sistema e identificarlos correctamente, así como realizar consultas de las horas de entrada y salidas.
<b>Objetivo</b>	OBJ-001
<b>Importancia</b>	Alta
<b>Estado</b>	Aprobado
<b>Estabilidad</b>	Alta
<b>Comentario</b>	Para la obtención del perfil del usuario se requiere interactuar con Active Directory.

<b>RQF-005</b>	<b>Registrar hora de entrada y salida extraordinaria</b>
<b>Descripción</b>	El sistema debe permitir al empleado ejecutar registros de salida extraordinarias.
<b>Objetivo</b>	OBJ-001 y OBJ-002
<b>Importancia</b>	Alta
<b>Estado</b>	Aprobado
<b>Estabilidad</b>	Alta
<b>Comentario</b>	Para la obtención del perfil del usuario se requiere interactuar con Active Directory.

## Requerimientos No Funcionales

<b>RQN-001</b>	<b>Imágenes en la interfaz de usuario</b>
<b>Descripción</b>	El sistema debe contar con una interfaz en la cual las imágenes no superen el 20% del tamaño del formulario Web y Windows.
<b>Objetivo</b>	NA
<b>Importancia</b>	Alta
<b>Estado</b>	Aprobado
<b>Estabilidad</b>	Alta
<b>Comentario</b>	NA

<b>Imagen</b>	NA
---------------	----

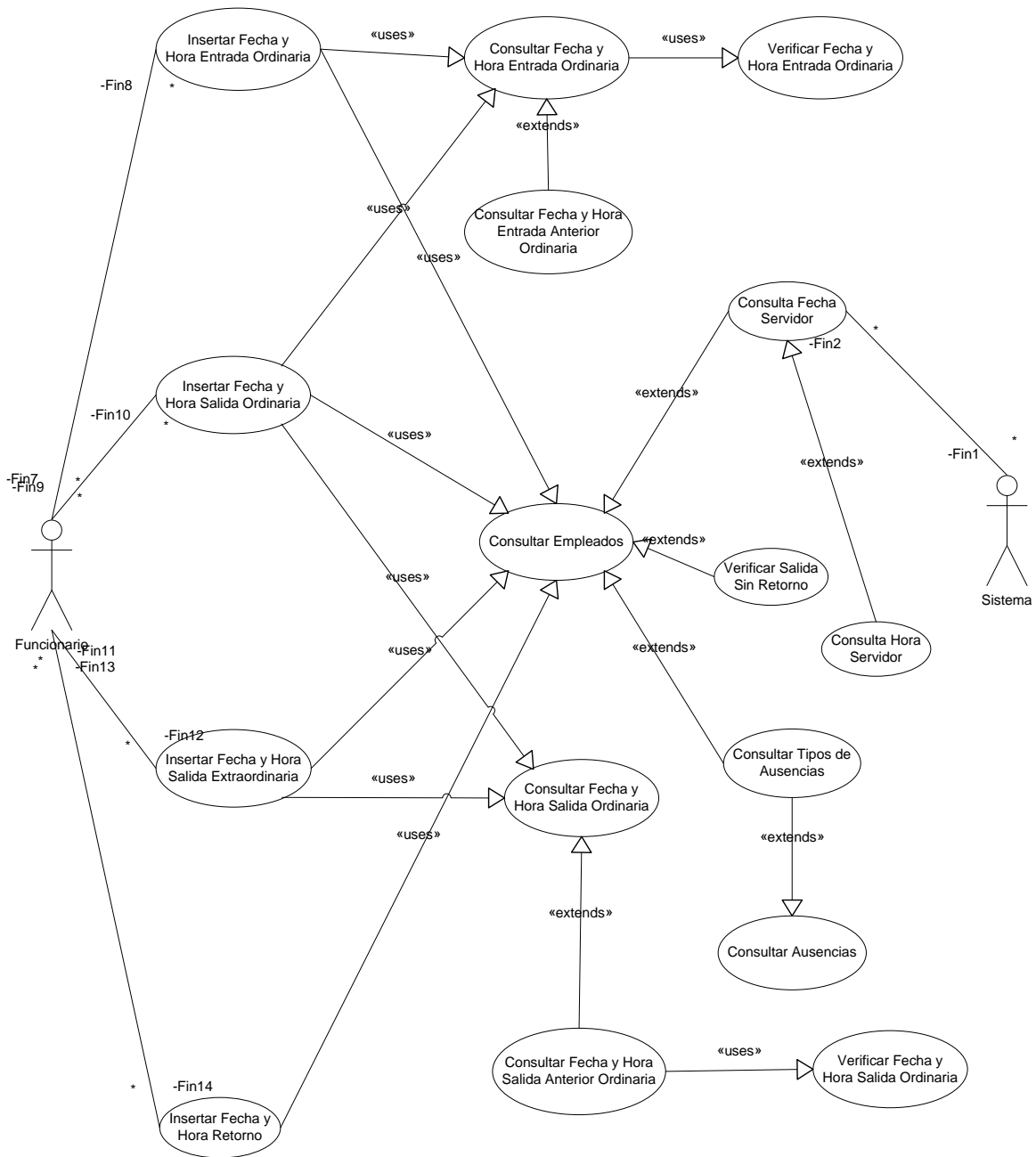
<b>RQN-002</b>	<b>Respuesta del sistema</b>
<b>Descripción</b>	El sistema debe responder ante cualquier transacción ejecutada por el usuario en un tiempo menor a los 5 segundos.
<b>Objetivo</b>	NA
<b>Importancia</b>	Alta
<b>Estado</b>	Aprobado
<b>Estabilidad</b>	Alta
<b>Comentario</b>	NA
<b>Imagen</b>	NA

<b>RQN-003</b>	<b>Eficiencia y respuesta ante fallos</b>
<b>Descripción</b>	El sistema debe poder recuperarse ante desconexiones o fallos de los servidores o de software. El sistema se debe mantener funcionando en todo momento.
<b>Objetivo</b>	NA
<b>Importancia</b>	Alta
<b>Estado</b>	Aprobado
<b>Estabilidad</b>	Alta
<b>Comentario</b>	NA
<b>Imagen</b>	NA

## Requerimiento de Rendimiento

<b>RQR-001</b>	<b>Cantidad de terminales</b>
<b>Descripción</b>	Se espera que el sistema trabaje en 30 terminales o más.
<b>Objetivo</b>	NA
<b>Importancia</b>	Alta
<b>Estado</b>	Aprobado
<b>Estabilidad</b>	Media
<b>Comentario</b>	NA

# Diagramas General de Casos de Uso



## Casos de Uso de Alto Nivel

<b>CU-001</b>	<b>Insertar fecha y hora entrada ordinaria</b>
<b>Requerimientos</b>	RQF-001, RQF-002
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001
<b>Tipo</b>	Primario
<b>Descripción</b>	El usuario identificado registra la fecha y hora de entrada ordinaria.

<b>CU-002</b>	<b>Insertar fecha y hora salida ordinaria</b>
<b>Requerimientos</b>	RQF-001, RQF-002
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001
<b>Tipo</b>	Primario
<b>Descripción</b>	El usuario es identificado registra la fecha y hora de la salida ordinaria.

<b>CU-003</b>	<b>Insertar fecha y hora salida extraordinaria</b>
<b>Requerimientos</b>	RQF-001, RQF-002
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001
<b>Tipo</b>	Primario
<b>Descripción</b>	El usuario es identificado registra la fecha y hora de la salida extraordinaria.

<b>CU-004</b>	<b>Insertar fecha y hora retorno</b>
<b>Requerimientos</b>	RQF-001, RQF-002
<b>Casos de Uso</b>	NA

<b>Relacionados</b>	
<b>Actores</b>	ACT-001
<b>Tipo</b>	Primario
<b>Descripción</b>	El usuario identificado registra la fecha y hora de retorno ya sea después del almuerzo o al inicio del día (al iniciar su jornada diaria de labores).

<b>CU-005</b>	<b>Consultar fecha y hora entrada ordinaria</b>
<b>Requerimientos</b>	RQF-001
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001, ACT-002
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema debe consultar si el usuarios al registrar una entrada ordinaria ya había registrado una entrada ordinaria durante ese día.

<b>CU-006</b>	<b>Consultar fecha y hora entrada anterior ordinaria</b>
<b>Requerimientos</b>	RQF-001
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001, ACT-002
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema obtiene la última la fecha y hora de entrada disponible para al usuario que esta registrando una entrada.

<b>CU-007</b>	<b>Verificar fecha y hora entrada ordinaria</b>
<b>Requerimientos</b>	RQF-001
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001, ACT-002
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema valida que la fecha y hora de la entrada ordinaria

	ya haya sido registrada cuando el usuario se dispone a registrar una entrada ordinaria.
--	---

<b>CU-008</b>	<b>Consultar empleado</b>
<b>Requerimientos</b>	RQF-004
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001, ACT-002
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema obtiene los datos del empleado registrado en la base de datos de Babel.

<b>CU-009</b>	<b>Verificar salida sin retorno</b>
<b>Requerimientos</b>	RQF-001
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema debe alertar al usuario cuando exista una salida sin retorno.

<b>CU-0010</b>	<b>Consultar fecha servidor</b>
<b>Requerimientos</b>	RQF-001
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-002
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema obtiene la fecha del servidor.

<b>CU-0011</b>	<b>Consultar hora servidor</b>
<b>Requerimientos</b>	RQF-001
<b>Casos de Uso Relacionados</b>	NA

<b>Actores</b>	ACT-002
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema obtiene la hora del servidor.

<b>CU-0012</b>	<b>Consultar tipo de ausencias</b>
<b>Requerimientos</b>	RQF-003
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema obtiene un listado de los tipos de ausencia disponibles dentro del catálogo.

<b>CU-0013</b>	<b>Consultar ausencias</b>
<b>Requerimientos</b>	RQF-003
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001
<b>Tipo</b>	Primario
<b>Descripción</b>	El usuario es puede consultar sus ausencias.

<b>CU-0014</b>	<b>Consultar fecha y hora salida ordinaria</b>
<b>Requerimientos</b>	RQF-001, RQF-002
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema debe consultar si el usuarios al registrar una salida ordinaria ya había registrado una salida ordinaria durante ese día.

<b>CU-0015</b>	<b>Consultar fecha y hora salida anterior ordinaria</b>
<b>Requerimientos</b>	RQF-005

<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema obtiene la última la fecha y hora de salida disponible para al usuario que esta registrando una salida.

<b>CU-0016</b>	<b>Verificar fecha y hora salida ordinaria</b>
<b>Requerimientos</b>	RQF-005
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema valida que la fecha y hora de la salida ordinaria ya haya sido registrada cuando el usuario se dispone a registrar una salida ordinaria.



## 4.2 PLANEACIÓN

### 4.2.1 PLAN DE TOMA DE REQUISITOS



# Plan de Toma de Requerimientos

---

**REGASIS**

**Versión 1.0.0**

**Diana Carolina Reyes Reyes**

**2008-07-14**

## **Función de la Plantilla**

La función de esta plantilla es estandarizar y formalizar todos los aspectos dentro del Plan de Toma de Requerimientos para los proyectos.

## **Objetivos de la Plantilla**

1. Estandarizar el Plan de Toma de Requerimientos Inicial.
2. Crear una estructura para la declaración de trabajo.

## **Alcance de la Plantilla**

Esta plantilla sólo se usará cuando se va a ir donde el cliente por primera vez para obtener un marco macro del proyecto para poder hacer la propuesta.

## Revisiones de Plantilla

Versión	Fecha de Entrega	Revisado por
1.0.0	2007-11-07	Esteban Oviedo

## Control de versiones Plantilla

Versión	Encargado	Descripción
1.0.0	Elías Conejo Cantillo	Creación de plantilla

## Revisiones

Versión	Fecha de Entrega	Revisado por
1.0.0	2007-09-13	Michael Bogantes

## Control de versiones

Versión	Encargado	Descripción
1.0.0	Diana Carolina Reyes Reyes	Creación de documento

## Carta de Aprobación

Por medio de la firma de esta carta de aceptación se aprueba la información del plan de toma de requerimientos inicial.

---

**Ing. Esteban Oviedo Blanco**

Director de Proyectos / Chief Projects Officer

## Tabla de contenido

<b>Función de la Plantilla .....</b>	<b>- 86 -</b>
<b>Objetivos de la Plantilla .....</b>	<b>- 86 -</b>
<b>Alcance de la Plantilla.....</b>	<b>- 86 -</b>
<b>Revisiones de Plantilla .....</b>	<b>- 87 -</b>
<b>Control de versiones Plantilla .....</b>	<b>- 87 -</b>
<b>Revisiones .....</b>	<b>- 87 -</b>
<b>Control de versiones.....</b>	<b>- 87 -</b>
<b>Carta de Aprobación .....</b>	<b>- 87 -</b>
<b>Tabla de contenido.....</b>	<b>- 88 -</b>
<b>Entrevistados .....</b>	<b>- 89 -</b>

## Entrevistados

<b>Nombre</b>	Esteban Oviedo
<b>Empresa</b>	Babel Software
<b>Puesto</b>	Gerente de Proyectos
<b>Objetivos</b>	<ol style="list-style-type: none"><li>1. Describir el panorama general del proyecto</li><li>2. Enumerar las funcionalidades generales del sistema a realizar</li><li>3. Conocer las expectativas del cliente hacia el producto a realizar</li></ol>
<b>Entrevista</b>	crm.ini.ent.1.REGASIS.EOB.doc.doc
<b>Lugar</b>	Sala de Juntas, Babel Software
<b>Fecha</b>	Lunes 12, de abril de 2008
<b>Hora</b>	9:00 a.m.

<b>Nombre</b>	Sady Valerio
<b>Empresa</b>	Babel Software
<b>Puesto</b>	Gerente de Recursos Humanos
<b>Objetivos</b>	<ol style="list-style-type: none"><li>1. Enumerar las funcionalidades del sistema</li><li>2. Conocer las expectativas del cliente hacia el producto a realizar</li></ol>
<b>Entrevista</b>	pjt.req.ent.01.REGASIS.SVV.doc
<b>Lugar</b>	Sala de Juntas, Babel Software
<b>Fecha</b>	Lunes 26, de mayo de 2008
<b>Hora</b>	9:00 a.m.

## 4.2.2 PLAN DE IMPLANTACIÓN



# Plan de Implantación

---

**REGASIS**

**Versión 1.0.0**

**Diana Carolina Reyes Reyes**

**2008-09-13**

## **Función de la Plantilla**

La función de esta plantilla es estandarizar y formalizar todos los aspectos dentro de un plan de implantación, ejemplo: los recursos necesarios por parte del cliente para que la implantación del sistema se haga de forma exitosa.

## **Objetivos de la Plantilla**

1. Estandarizar la especificación del plan de implantación.
2. Crear una estructura para el plan de implantación.

## **Alcance de la Plantilla**

Esta plantilla sólo se usará después de que termine un proyecto, antes de iniciar la implantación del sistema creado.

## Revisiones de Plantilla

Versión	Fecha de Entrega	Revisado por

## Control de versiones Plantilla

Versión	Encargado	Descripción
1.0.0	Elías Conejo Cantillo	Creación de plantilla

## Revisiones

Versión	Fecha de Entrega	Revisado por
1.0.0	2007-09-13	Michael Bogantes

## Control de versiones

Versión	Encargado	Descripción
1.0.0	Diana Carolina Reyes Reyes	Creación de documento

## Carta de Aprobación

Por medio de la firma de esta carta de aceptación se establece que el plan de implantación fue revisado y pasó por un proceso de aseguramiento de la calidad. Además el cliente se compromete a tener los recursos necesarios (personal, equipo, contraseñas, etc.) para la fecha que el plan indica, a la hora que el plan indica y por el tiempo que éste lo indica. Cualquier desacato a este plan por parte del cliente provocará atrasos y por lo tanto un **costo adicional** para el cliente.

---

**Ing. Esteban Oviedo Blanco**

Director de Proyectos / Chief Projects Officer



## Implantación

<b>Acción</b>	RPD-001
<b>Nombre</b>	Instalación en el Servidor
<b>Descripción</b>	<ol style="list-style-type: none"> <li>1. Instalar Framework .Net v2.0</li> <li>2. Instalar Ajax</li> <li>3. Instalar la Base de Datos SQL Server 2005</li> <li>4. Instalar Reporting Services</li> <li>5. Instalar los Servicios Web</li> <li>6. Configuración de la aplicación de Servios Web</li> </ol>
<b>Fecha</b>	2008-20-01
<b>Hora</b>	8:00 a.m.
<b>Se necesita ayuda del personal del cliente</b>	Sí
<b>Nombre de la persona(s) por parte del cliente que se necesitan para la acción</b>	Diana Carolina Reyes Reyes
<b>Datos necesarios</b>	Contraseñas del servidor y permisos para instalación
<b>Tiempo estimado</b>	4 horas

<b>Acción</b>	RPD-002
<b>Nombre</b>	Instalación en cliente
<b>Descripción</b>	<ol style="list-style-type: none"> <li>1. Instalar Framework .Net v2.0</li> <li>2. Instalar Actualizaciones de Internet Explorer</li> <li>3. Instalar Aplicación "REGASIS"</li> <li>4. Configurar el acceso al servidor de Reporting Services</li> <li>5. Configurar el acceso a los Servicios Web</li> <li>6. Configuración de la aplicación "REGASIS"</li> </ol>
<b>Fecha</b>	2008-20-01
<b>Hora</b>	1:00 p.m.

<b>Se necesita ayuda del personal del cliente</b>	Sí
<b>Nombre de la persona(s) por parte del cliente que se necesitan para la acción</b>	Diana Carolina Reyes Reyes
<b>Datos necesarios</b>	Cuentas de Active Directory
<b>Tiempo estimado</b>	4 horas

## 4.3 EJECUCIÓN

### 4.3.1 ESPECIFICACIÓN DE DISEÑO



# Especificación de Diseño

---

REGASIS

Versión 1.0.0

Diana Carolina Reyes Reyes

2008-06-01

## **Función de la Plantilla**

La función de esta plantilla es estandarizar y formalizar todos los aspectos dentro de una especificación de requerimientos, ejemplo: casos de uso. Para esto se utilizarán estándares mundiales (IEEE, UML).

## **Objetivos de la Plantilla**

1. Estandarizar la especificación del diseño del sistema.
2. Especificar los casos de uso con el estándar UML.
3. Especificar las interacciones entre capas con el estándar UML.
4. Crear una estructura para la especificación de diseño.
5. Ayudar en el rastreo de requerimientos.

## **Alcance de la Plantilla**

Esta plantilla sólo se usará después de la toma de requerimientos, después de la creación de la ERS, cuando se tenga que especificar qué forma tendrá el sistema a crear y bajo qué arquitectura se desarrollará.

## Revisiones de Plantilla

Versión	Fecha de Entrega	Revisado por
1.0.0	2007-12-21	Esteban Oviedo

## Control de versiones Plantilla

Versión	Encargado	Descripción
1.0.0	Elías Conejo Cantillo	Creación de plantilla

## Revisiones

Versión	Fecha de Entrega	Revisado por
1.0.0	2007-09-13	Michael Bogantes

## Control de versiones

Versión	Encargado	Descripción
1.0.0	Diana Carolina Reyes Reyes	Creación de documento

## Carta de Aprobación

Por medio de la firma de esta carta de aceptación los casos de uso aquí descritos se aprueban y serán desarrollados. Se acepta que no quedan casos de uso, correspondientes a los requerimientos solicitados fuera del alcance del proyecto.

---

**Ing. Esteban Oviedo Blanco**

Director de Proyectos / Chief Projects Officer

## Tabla de contenido

<b>Función de la Plantilla .....</b>	<b>- 96 -</b>
<b>Objetivos de la Plantilla.....</b>	<b>- 96 -</b>
<b>Alcance de la Plantilla.....</b>	<b>- 96 -</b>
<b>Revisiones de Plantilla .....</b>	<b>- 97 -</b>
<b>Control de versiones Plantilla.....</b>	<b>- 97 -</b>
<b>Revisiones .....</b>	<b>- 97 -</b>
<b>Control de versiones.....</b>	<b>- 97 -</b>
<b>Carta de Aprobación .....</b>	<b>- 97 -</b>
<b>Tabla de contenido.....</b>	<b>- 98 -</b>
<b>Introducción .....</b>	<b>- 102 -</b>
Propósito .....	- 102 -
Definiciones, Acrónimos y Abreviaturas .....	- 102 -
Visión General del Documento .....	- 102 -
<b>Diagrama de Componentes.....</b>	<b>- 103 -</b>
<b>Diagrama de Despliegue.....</b>	<b>- 104 -</b>
<b>Diagrama de Clases .....</b>	<b>- 105 -</b>
<b>Definición de casos de uso en formato expandido.....</b>	<b>- 106 -</b>
Insertar fecha y hora entrada ordinaria.....	- 106 -
Insertar fecha y hora salida ordinaria.....	- 106 -
Insertar fecha y hora salida extraordinaria.....	- 107 -
Insertar fecha y hora de retorno .....	- 108 -
Consultar fecha y hora entrada ordinaria.....	- 109 -
Consultar fecha y hora entrada anterior ordinaria .....	- 110 -
Verificar fecha y hora entrada ordinaria.....	- 110 -
Consultar empleado.....	- 111 -
Verificar salida sin retorno .....	- 112 -
Consultar fecha servidor.....	- 113 -
Consultar hora servidor .....	- 114 -
Consultar tipo de ausencias .....	- 114 -
Consultar ausencias .....	- 115 -
Consultar fecha y hora salida ordinaria .....	- 116 -

Consultar fecha y hora salida anterior ordinaria .....	- 117 -
Verificar fecha y hora salida ordinaria.....	- 118 -
<b>Arquitectura del Sistema .....</b>	<b>- 119 -</b>
Capa de Presentación / Interfaz de usuario.....	- 119 -
Capa de Servicios.....	- 119 -
Capa Lógica .....	- 119 -
Capa de Acceso de Datos .....	- 119 -
<b>Diagramas de Interacción.....</b>	<b>- 120 -</b>
CU-001: Insertar fecha y hora entrada ordinaria .....	- 120 -
CU-002: Insertar fecha y hora salida ordinaria .....	- 120 -
CU-003: Insertar fecha y hora salida extraordinaria .....	- 121 -
CU-004: Insertar fecha y hora retorno .....	- 121 -
CU-005: Consultar fecha y hora entrada ordinaria .....	- 121 -
CU-006: Consultar fecha y hora entrada anterior ordinaria .....	- 122 -
CU-007: Verificar fecha y hora entrada ordinaria .....	- 122 -
CU-008: Consultar empleado .....	- 122 -
CU-009: Verificar salida sin retorno .....	- 123 -
CU-010: Consultar fecha servidor .....	- 123 -
CU-011: Consultar hora servidor .....	- 123 -
CU-012: Consultar tipo de ausencias .....	- 124 -
CU-013: Consultar ausencias.....	- 124 -
CU-014: Consultar fecha y hora salida ordinaria .....	- 124 -
CU-015: Consultar fecha y hora salida anterior ordinaria .....	- 125 -
CU-016: Verificar fecha y hora salida ordinaria .....	- 125 -
<b>Diagramas de Colaboración.....</b>	<b>- 126 -</b>
CU-001: Insertar fecha y hora entrada ordinaria .....	- 126 -
CU-002: Insertar fecha y hora salida ordinaria .....	- 126 -
CU-003: Insertar fecha y hora salida extraordinaria .....	- 127 -
CU-004: Insertar fecha y hora retorno .....	- 127 -
CU-005: Consultar fecha y hora entrada ordinaria .....	- 128 -
CU-006: Consultar fecha y hora entrada anterior ordinaria .....	- 128 -
CU-007: Verificar fecha y hora entrada ordinaria .....	- 129 -
CU-008: Consultar empleado .....	- 129 -
CU-009: Verificar salida sin retorno .....	- 130 -

CU-010: Consultar fecha servidor .....	- 130 -
CU-011: Consultar hora servidor .....	- 131 -
CU-012: Consultar tipo de ausencias .....	- 131 -
CU-013: Consultar ausencias .....	- 132 -
CU-014: Consultar fecha y hora salida ordinaria .....	- 132 -
CU-015: Consultar fecha y hora salida anterior ordinaria .....	- 133 -
CU-016: Verificar fecha y hora salida ordinaria .....	- 133 -
<b>Diagramas de Estado .....</b>	<b>- 134 -</b>
CU-001: Insertar fecha y hora entrada ordinaria .....	- 134 -
CU-002: Insertar fecha y hora salida ordinaria .....	- 134 -
CU-003: Insertar fecha y hora salida extraordinaria .....	- 135 -
CU-004: Insertar fecha y hora retorno .....	- 135 -
CU-005: Consultar fecha y hora entrada ordinaria .....	- 136 -
CU-006: Consultar fecha y hora entrada anterior ordinaria .....	- 136 -
CU-007: Verificar fecha y hora entrada ordinaria .....	- 137 -
CU-008: Consultar empleado .....	- 137 -
CU-009: Verificar salida sin retorno .....	- 138 -
CU-010: Consultar fecha servidor .....	- 138 -
CU-011: Consultar hora servidor .....	- 139 -
CU-012: Consultar tipo de ausencias .....	- 139 -
CU-013: Consultar ausencias .....	- 140 -
CU-014: Consultar fecha y hora salida ordinaria .....	- 140 -
CU-015: Consultar fecha y hora salida anterior ordinaria .....	- 141 -
CU-016: Verificar fecha y hora salida ordinaria .....	- 141 -
<b>Diagramas de Actividad .....</b>	<b>- 142 -</b>
CU-001: Insertar fecha y hora entrada ordinaria .....	- 142 -
CU-002: Insertar fecha y hora salida ordinaria .....	- 142 -
CU-003: Insertar fecha y hora salida extraordinaria .....	- 143 -
CU-004: Insertar fecha y hora retorno .....	- 143 -
CU-005: Consultar fecha y hora entrada ordinaria .....	- 144 -
CU-006: Consultar fecha y hora entrada anterior ordinaria .....	- 144 -
CU-007: Verificar fecha y hora entrada ordinaria .....	- 145 -
CU-008: Consultar empleado .....	- 145 -
CU-009: Verificar salida sin retorno .....	- 146 -



CU-010: Consultar fecha servidor .....	- 146 -
CU-011: Consultar hora servidor .....	- 147 -
CU-012: Consultar tipo de ausencias .....	- 147 -
CU-013: Consultar ausencias .....	- 148 -
CU-014: Consultar fecha y hora salida ordinaria .....	- 148 -
CU-015: Consultar fecha y hora salida anterior ordinaria .....	- 149 -
CU-016: Verificar fecha y hora salida ordinaria .....	- 149 -

## **Introducción**

El presente documento especifica la estructura de diseño propuesta para el sistema REGASIS en el que se indican las especificaciones necesarias para tener claro el diseño y construcción de este software.

Se detalla la interacción entre las capas del software y sus diferentes componentes.

## **Propósito**

El propósito de este documento es servir como guía y ser una especificación del cómo debe ser desarrollado el software REGASIS.

El documento está dirigido a todos aquellos involucrados en el proyecto de construcción de este sistema, que incluyo a desarrolladores, administradores de proyecto y el patrocinador.

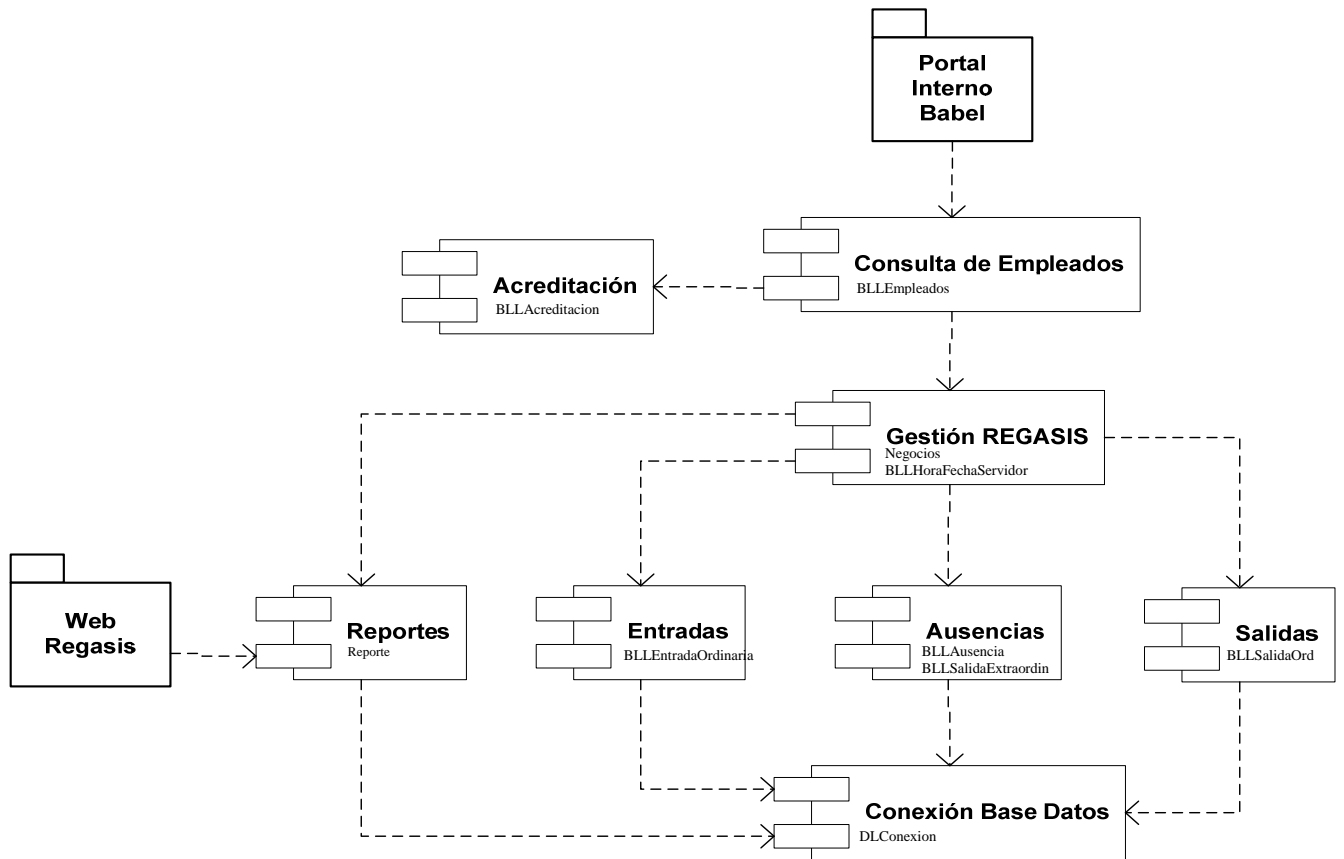
## **Definiciones, Acrónimos y Abreviaturas**

- **EDS:** Especificación de diseño.

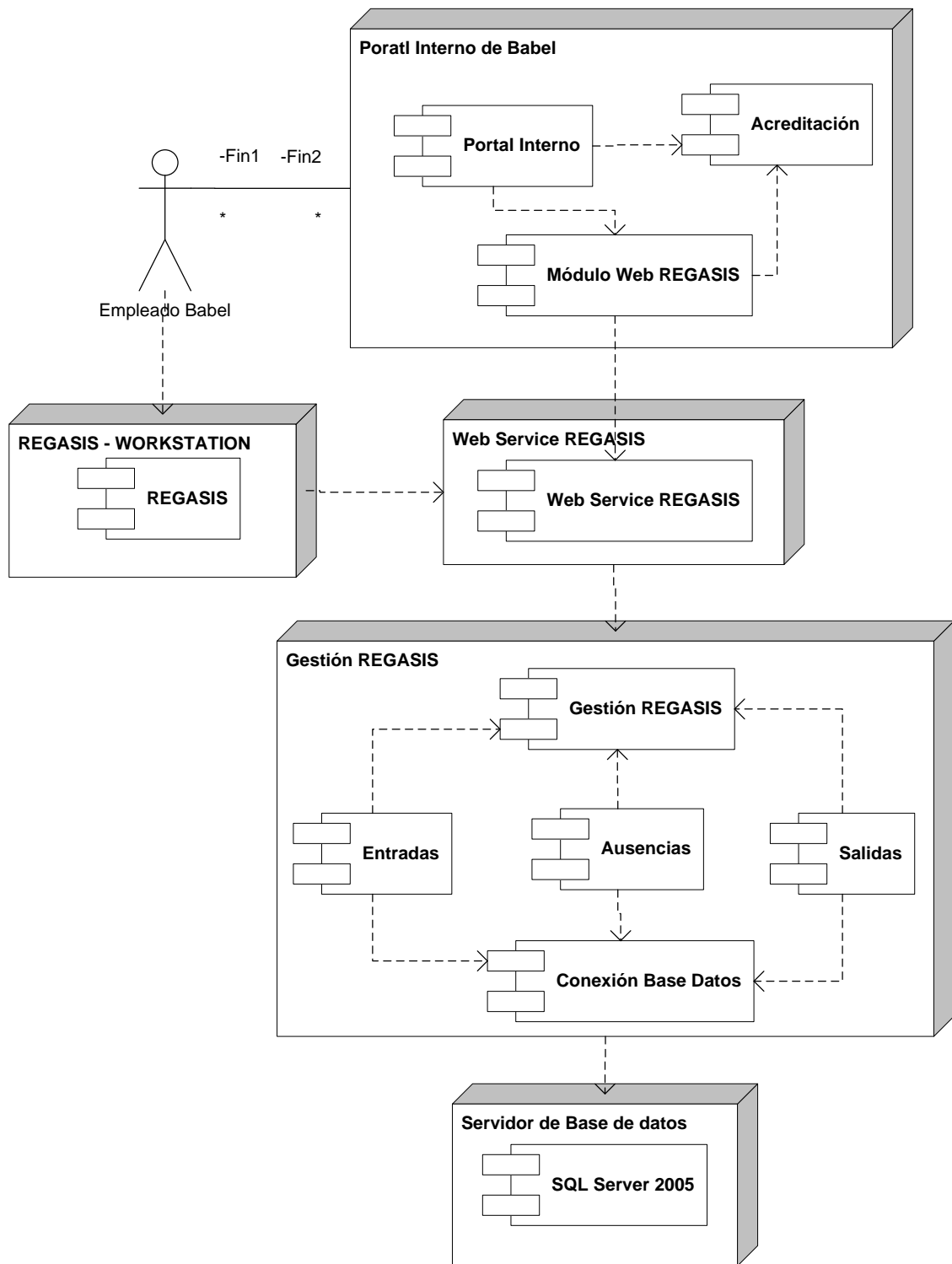
## **Visión General del Documento**

En las siguientes secciones se describirán las clases, así como se diagramarán los diferentes casos de uso del sistema.

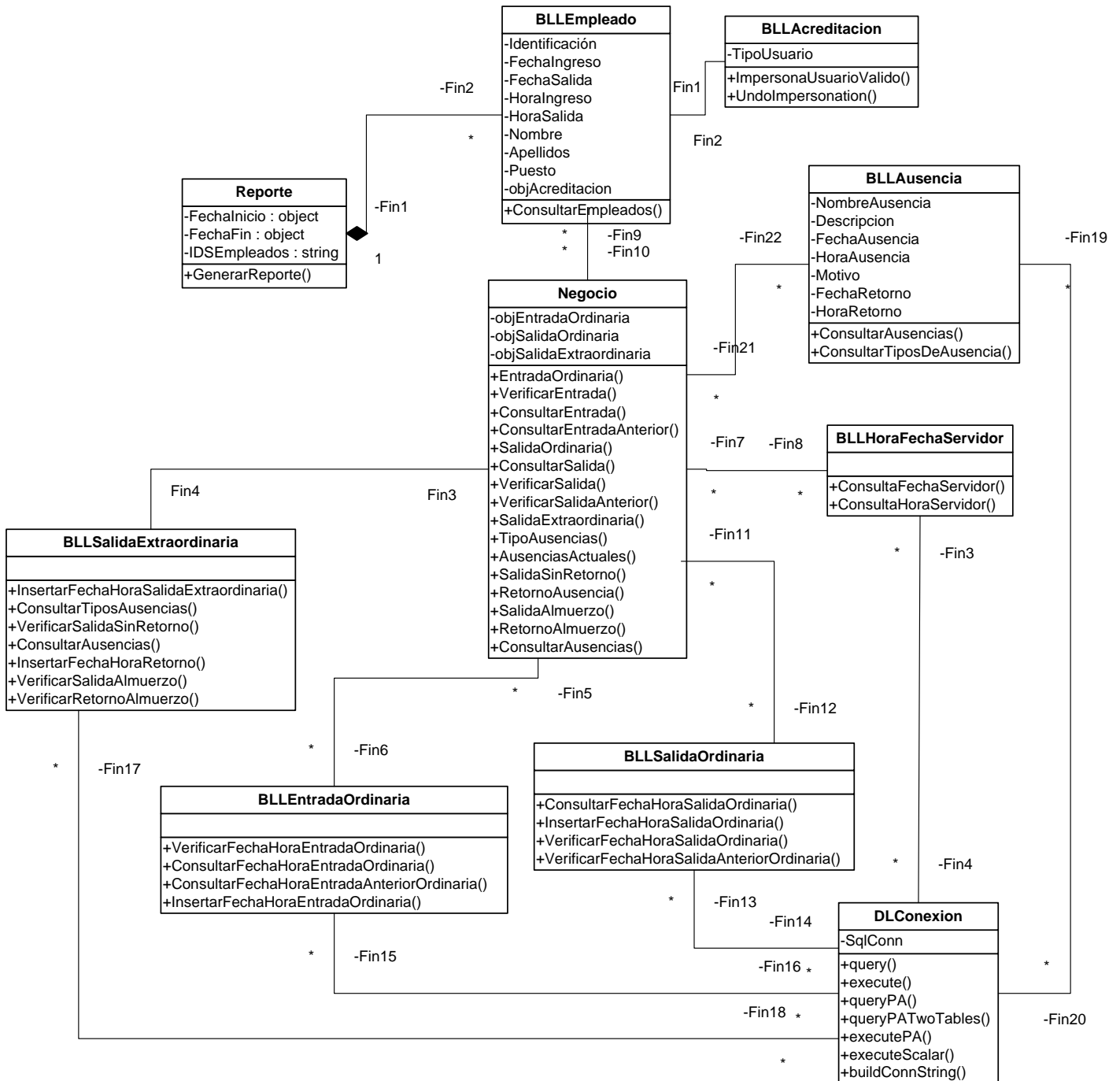
# Diagrama de Componentes



# Diagrama de Despliegue



# Diagrama de Clases



## Definición de casos de uso en formato expandido

<b>CU-001</b>	<b>Insertar fecha y hora entrada ordinaria</b>	
<b>Requerimientos</b>	RQF-001, RQF-002	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-001	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El usuario identificado registra la fecha y hora de entrada ordinaria.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	1. El sistema actualiza el estado del empleado	
<b>Curso Típico de Eventos</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta el rol del usuario.
	3. El actor oprime el botón "Marcar hora de entrada"	4. El sistema ingresa la hora de entrada.
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

<b>CU-002</b>	<b>Insertar fecha y hora salida ordinaria</b>	
<b>Requerimientos</b>	RQF-001, RQF-002	
<b>Casos de Uso Relacionados</b>	NA	

<b>Actores</b>	ACT-001	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El usuario es identificado registra la fecha y hora de la salida ordinaria.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	1. El sistema actualiza el estado del empleado	
<b>Curso Típico de Eventos</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta el rol del usuario.
	3. El actor oprime el botón "Marcar hora de salida".	4. El sistema ingresa la hora de salida.
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

<b>CU-003</b>	<b>Insertar fecha y hora salida extraordinaria</b>	
<b>Requerimientos</b>	RQF-001, RQF-002	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-001	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El usuario es identificado registra la fecha y hora de la salida extraordinaria.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	1. El sistema actualiza el estado del empleado	

Curso Típico de Eventos	Acción del Actor	Respuesta del Sistema
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta el rol del usuario.
	3. El actor oprime el botón "Marcar hora de salida"	4. El sistema ingresa la hora de salida extraordinaria.
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

<b>CU-004</b>	<b>Insertar fecha y hora de retorno</b>	
<b>Requerimientos</b>	RQF-001, RQF-002	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-001	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El usuario identificado registra la fecha y hora de retorno ya sea después del almuerzo o al inicio del día (al iniciar su jornada diaria de labores).	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	1. El sistema actualiza el estado del empleado	
Curso Típico de Eventos	Acción del Actor	Respuesta del Sistema
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta el rol del usuario
	3. El usuario oprime el botón "Registrar Retorno"	4. El sistema ingresa la hora de retorno.



	<b>Curso Alternativo:</b>
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.
<b>Importancia</b>	Alta
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.

<b>CU-005</b>	<b>Consultar fecha y hora entrada ordinaria</b>	
<b>Requerimientos</b>	RQF-001	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-001, ACT-002	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El sistema debe consultar si el usuarios al registrar una entrada ordinaria ya había registrado una entrada ordinaria durante ese día.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	NA	
<b>Curso Típico de Eventos</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta el rol del usuario.
	3. El actor selecciona la opción "Registro Diario"	4. El sistema muestra la hora de entrada que marco el usuario (Si la hay).
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
	Acción 4: Si no hubiera hora de entrada el sistema le pide al usuario que la ingrese.	
<b>Importancia</b>	Alta	

<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.
--------------------	--

<b>CU-006</b>	<b>Consultar fecha y hora entrada anterior ordinaria</b>	
<b>Requerimientos</b>	RQF-001	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-001, ACT-002	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El sistema obtiene la última la fecha y hora de entrada disponible para al usuario que está registrando una entrada.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	NA	
<b>Curso Típico de Eventos</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta el rol del usuario.
	3. El actor selecciona la opción "Registro Diario"	4. El sistema muestra la hora de entrada anterior ordinaria.
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
	Acción 4: Si no hubiera hora de entrada el sistema le pide al usuario que la ingrese.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

<b>CU-007</b>	<b>Verificar fecha y hora entrada ordinaria</b>
<b>Requerimientos</b>	RQF-001

<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-001, ACT-002	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El sistema valida que la fecha y hora de la entrada ordinaria ya haya sido registrada cuando el usuario se dispone a registrar una entrada ordinaria.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	NA	
<b>Curso Típico de Eventos</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta el rol del usuario.
	3. El actor selecciona la opción "Marcar hora de entrada"	4. El sistema valida si anteriormente el usuario no había registrado una hora de entrada.
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

<b>CU-008</b>	<b>Consultar empleado</b>
<b>Requerimientos</b>	RQF-004
<b>Casos de Uso Relacionados</b>	NA
<b>Actores</b>	ACT-001, ACT-002
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema obtiene los datos del empleado registrado en la

	base de datos de Babel.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	<b>NA</b>	
<b>Curso Típico de Eventos</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta el rol del usuario y el perfil del mismo mostrando su nombre en pantalla.
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

<b>CU-009</b>	<b>Verificar salida sin retorno</b>	
<b>Requerimientos</b>	RQF-001	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-001	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El sistema debe alertar al usuario cuando exista una salida sin retorno.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	<b>NA</b>	
<b>Curso Típico de Eventos</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta el rol del usuario y el perfil del mismo mostrando su nombre en pantalla.

	3. El usuario selecciona "Marcar hora de salida"	4. El sistema verifica si se había marcado antes la hora de entrada.
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

<b>CU-010</b>	<b>Consultar fecha servidor</b>	
<b>Requerimientos</b>	RQF-001	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-002	
<b>Tipo</b>	Primario	
<b>Descripción</b>	<b>NA</b>	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	1. El sistema permite al usuario el ingreso de una hora de entrada laboral y salida.	
<b>Curso Típico de Eventos</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta el rol del usuario y realiza una consulta al servidor para obtener la fecha en el formato (DD/MM/AAAA).
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

<b>CU-011</b>	<b>Consultar hora servidor</b>	
<b>Requerimientos</b>	RQF-001	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-002	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El sistema obtiene la fecha del servidor.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	NA	
<b>Curso Típico de Eventos</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta el rol del usuario y realiza una consulta al servidor para obtener la hora en el formato (HH:MM:SS AM/PM).
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

<b>CU-012</b>	<b>Consultar tipo de ausencias</b>	
<b>Requerimientos</b>	RQF-003	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-001	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El sistema obtiene un listado de los tipos de ausencia disponibles dentro del catálogo.	

<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	NA	
<b>Curso Típico de Eventos</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta el rol del usuario y realiza una consulta la descripción de las ausencias cargando un comboBox con las mismas.
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

<b>CU-013</b>	<b>Consultar ausencias</b>	
<b>Requerimientos</b>	RQF-003	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-001	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El usuario es puede consultar sus ausencias.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	NA	
<b>Curso Típico de</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

<b>Eventos</b>	<p>1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.</p>	<p>2. El módulo consulta el rol del usuario y realiza una consulta todas las ausencias que tiene el usuario durante ese día, morreando los siguiente datos:</p> <ul style="list-style-type: none"> <li>- Hora salida</li> <li>- Hora retorno</li> <li>- Tipo de ausencia</li> <li>- Motivo</li> </ul>
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

<b>CU-014</b>	<b>Consultar fecha y hora salida ordinaria</b>	
<b>Requerimientos</b>	RQF-001, RQF-002	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-001	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El sistema debe consultar si el usuarios al registrar una salida ordinaria ya había registrado una salida ordinaria durante ese día.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	NA	
<b>Curso Típico de</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>



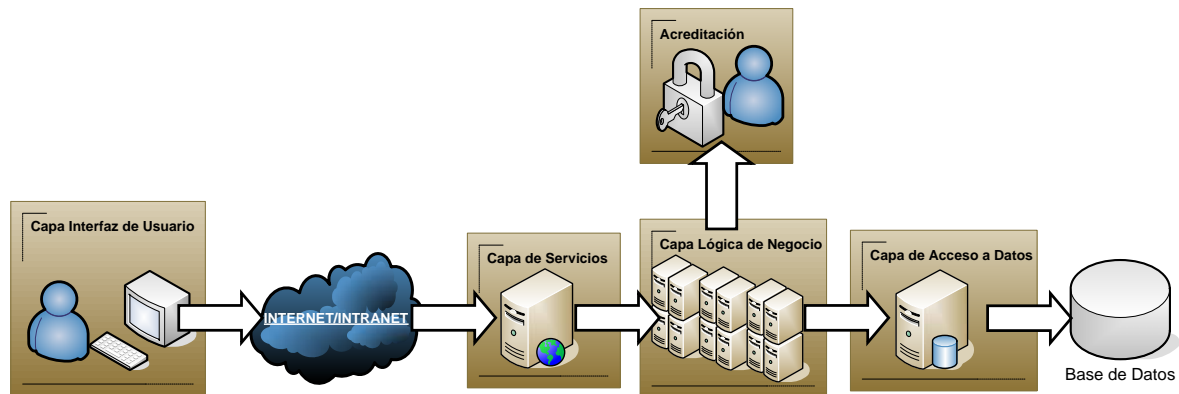
<b>Eventos</b>	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta la fecha y hora de salida del empleado.
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

<b>CU-015</b>	<b>Consultar fecha y hora salida anterior ordinaria</b>	
<b>Requerimientos</b>	RQF-005	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-001	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El sistema debe consultar si el usuarios al registrar una salida ordinaria ya había registrado una salida ordinaria durante ese día.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	<b>NA</b>	
<b>Curso Típico de Eventos</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta la fecha y hora de salida anterior del empleado.
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	

<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.
--------------------	--

<b>CU-016</b>	<b>Verificar fecha y hora salida ordinaria</b>	
<b>Requerimientos</b>	RQF-005	
<b>Casos de Uso Relacionados</b>	NA	
<b>Actores</b>	ACT-001	
<b>Tipo</b>	Primario	
<b>Descripción</b>	El sistema valida que la fecha y hora de la salida ordinaria ya haya sido registrada cuando el usuario se dispone a registrar una salida ordinaria.	
<b>Precondiciones</b>	1. El usuario debe estar autenticado y validado desde el sistema de Seguridad.	
<b>Postcondiciones</b>	NA	
<b>Curso Típico de Eventos</b>	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso inicia cuando el Usuario accede al formulario principal del módulo.	2. El módulo consulta la fecha y hora de salida anterior ordinaria del empleado.
	<b>Curso Alternativo:</b>	
	Acción 1: El Usuario cancela la operación. El sistema no realiza ninguna operación.	
<b>Importancia</b>	Alta	
<b>Comentarios</b>	Esta funcionalidad es de suma importancia puesto que al módulo deben ingresar solo usuarios autorizados.	

# Arquitectura del Sistema



**Figura 4.2 Arquitectura del Sistema**

La arquitectura aplicada para el desarrollo del sistema es la de “N-Capas” o “Multicapas”.

## Capa de Presentación / Interfaz de usuario

Son todas las páginas Web o pantallas Windows que participan en la aplicación, desde aquí se ingresan los datos por medio de formularios.

## Capa de Servicios

Constituye un conjunto de Servicios Web, que funcionan como una interfaz de comunicación con la capa lógica; esto con el objeto de reforzar

## Capa Lógica

Es un conjunto de clases que realizan el proceso lógico y matemático de la aplicación, así como también el almacenamiento temporal de la información.

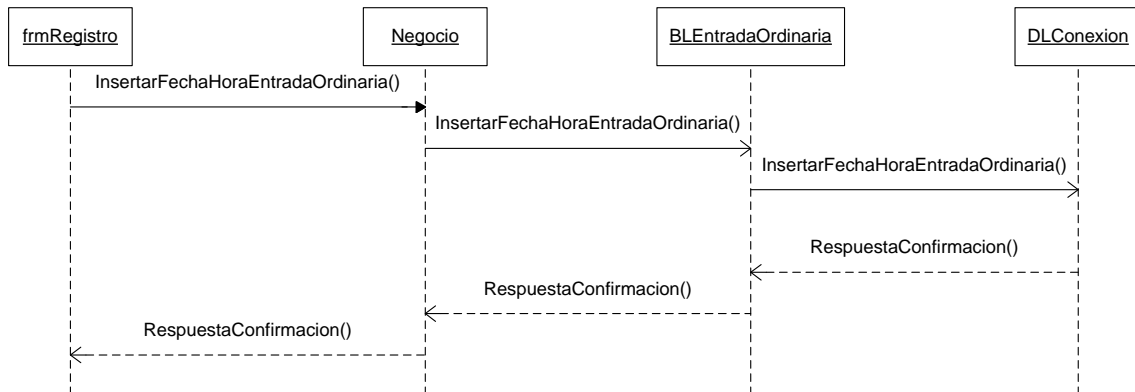
La aplicación utiliza la clase Acreditación para realizar la “impersonación” del usuario frente al Active Directory del dominio interno de Babel Software.

## Capa de Acceso de Datos

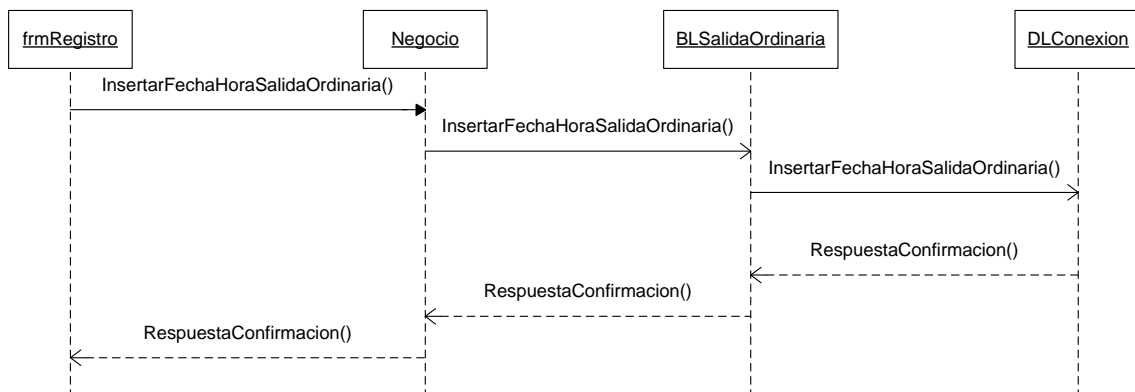
En esta capa se realiza la obtención y almacenamiento de los datos por medio del Controlador de Datos, la aplicación utiliza la Base de Datos es BabelSys y trabaja con la herramienta SQL Server.

## Diagramas de Interacción

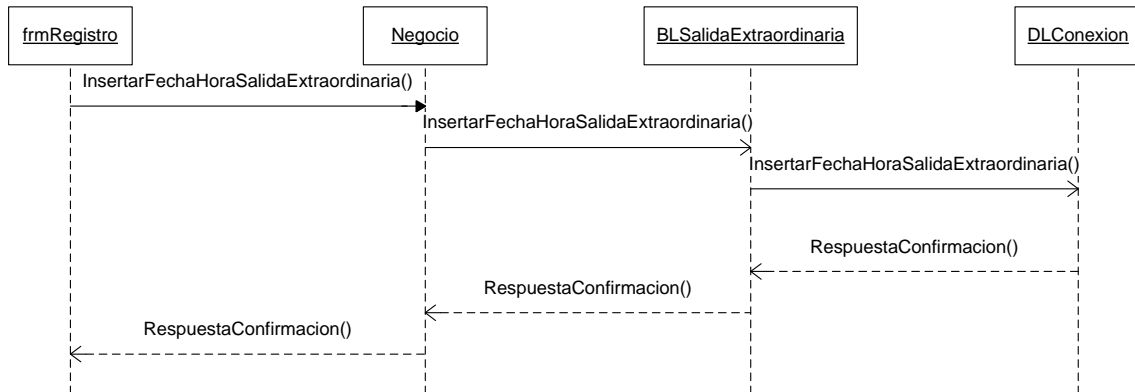
### CU-001: Insertar fecha y hora entrada ordinaria



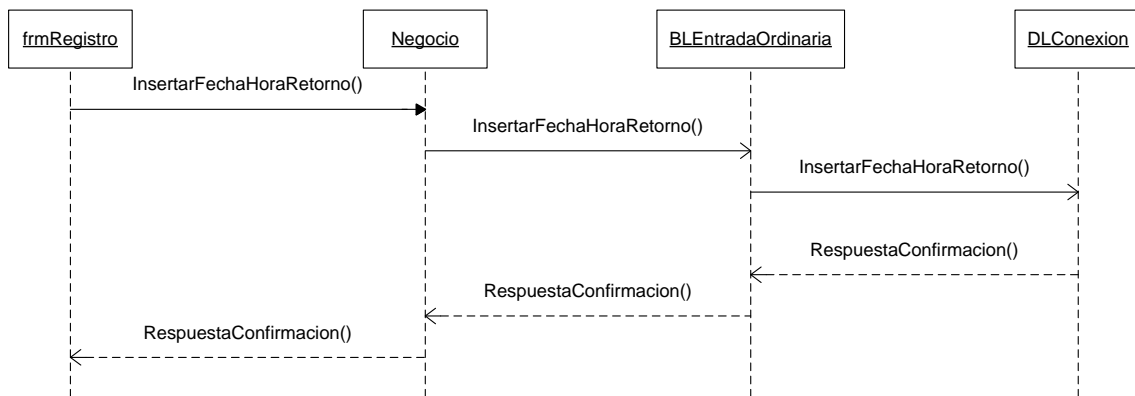
### CU-002: Insertar fecha y hora salida ordinaria



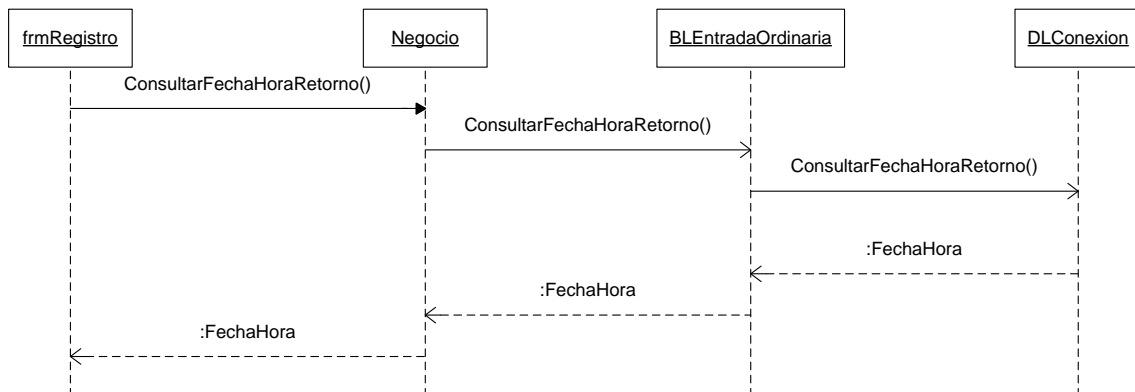
### CU-003: Insertar fecha y hora salida extraordinaria



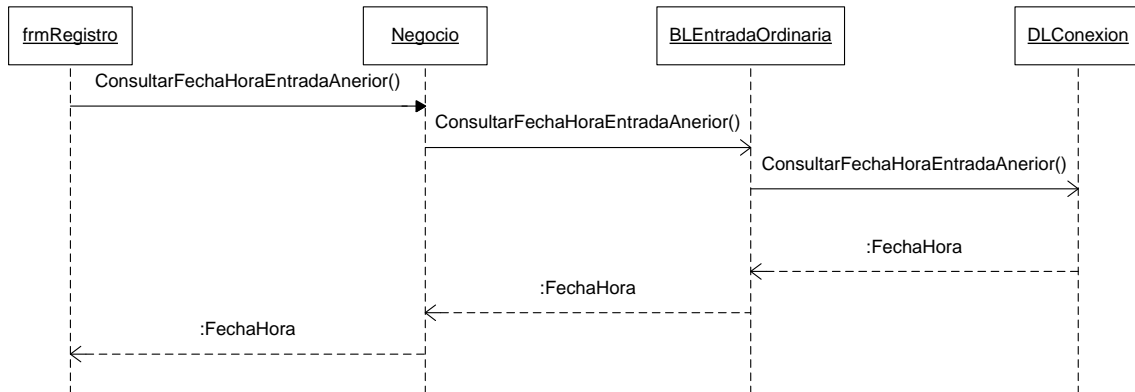
### CU-004: Insertar fecha y hora retorno



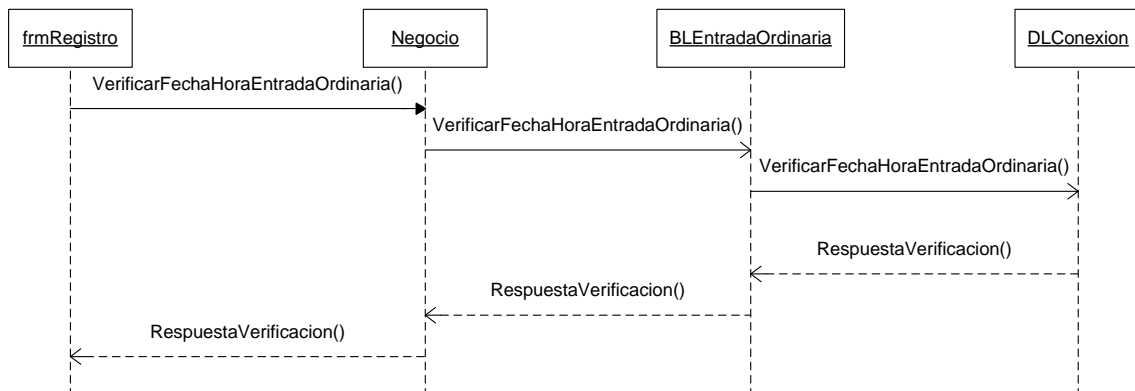
### CU-005: Consultar fecha y hora entrada ordinaria



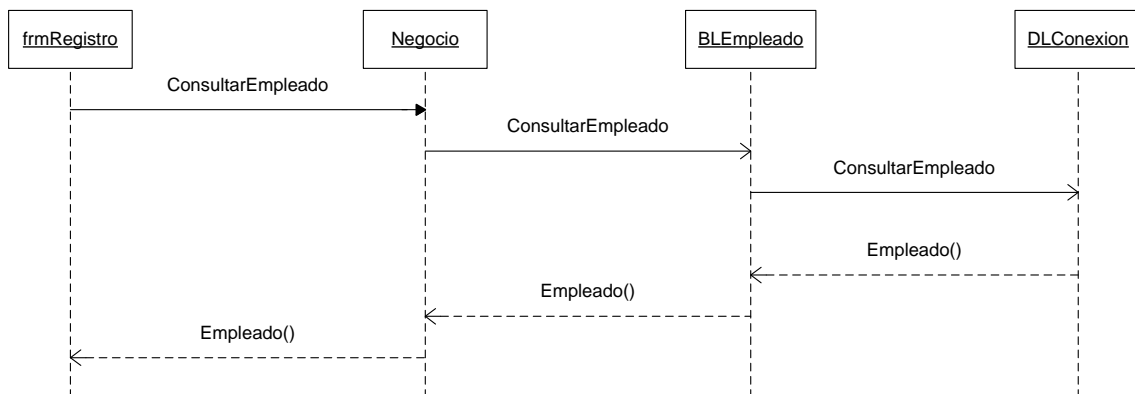
## CU-006: Consultar fecha y hora entrada anterior ordinaria



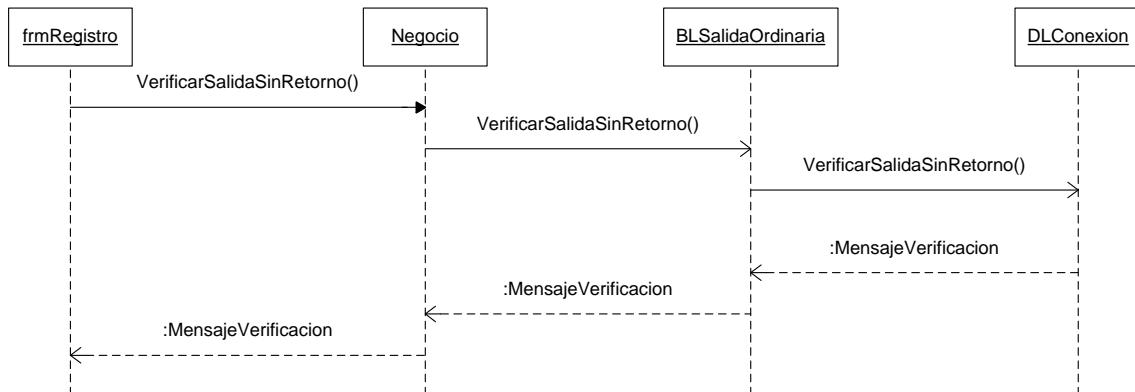
## CU-007: Verificar fecha y hora entrada ordinaria



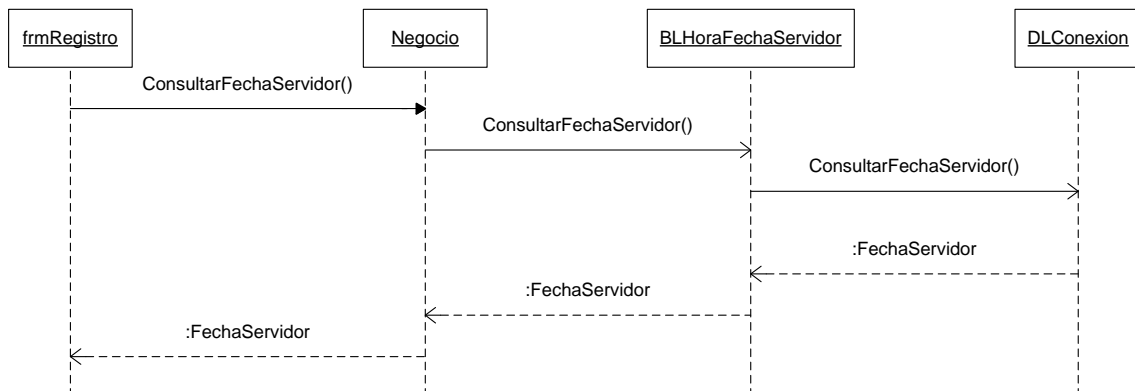
## CU-008: Consultar empleado



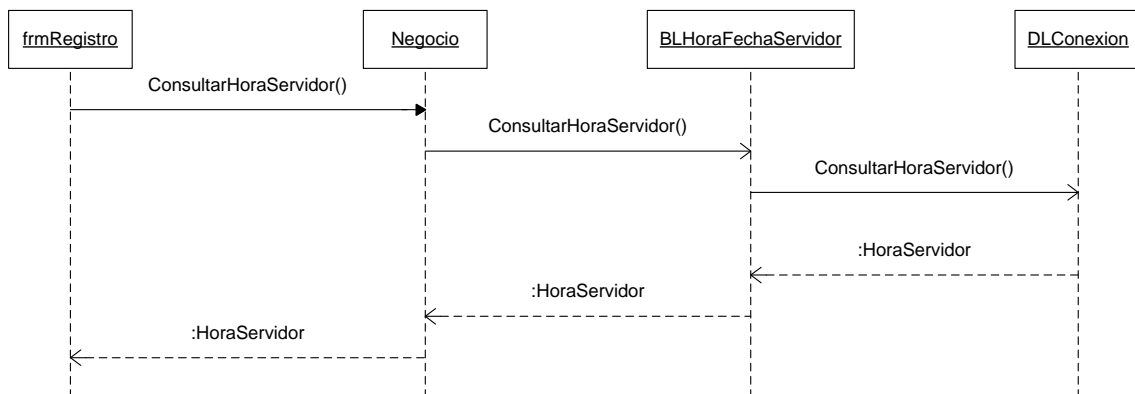
## CU-009: Verificar salida sin retorno



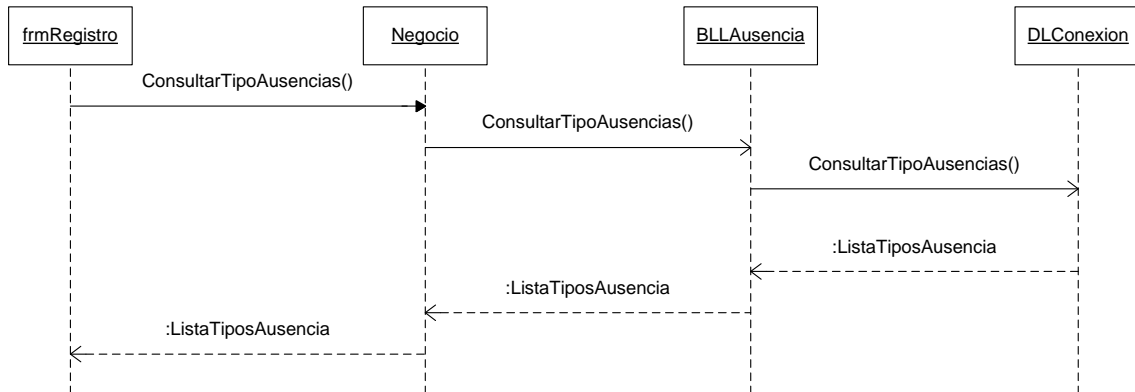
## CU-010: Consultar fecha servidor



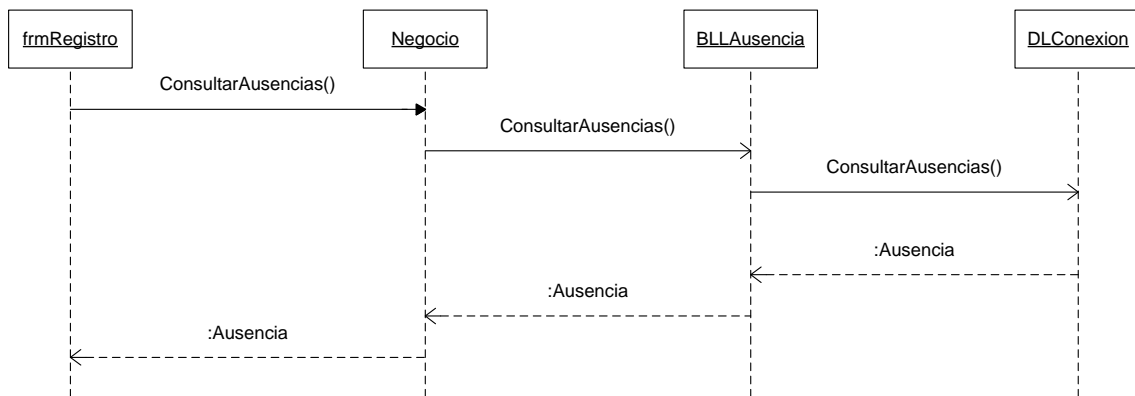
## CU-011: Consultar hora servidor



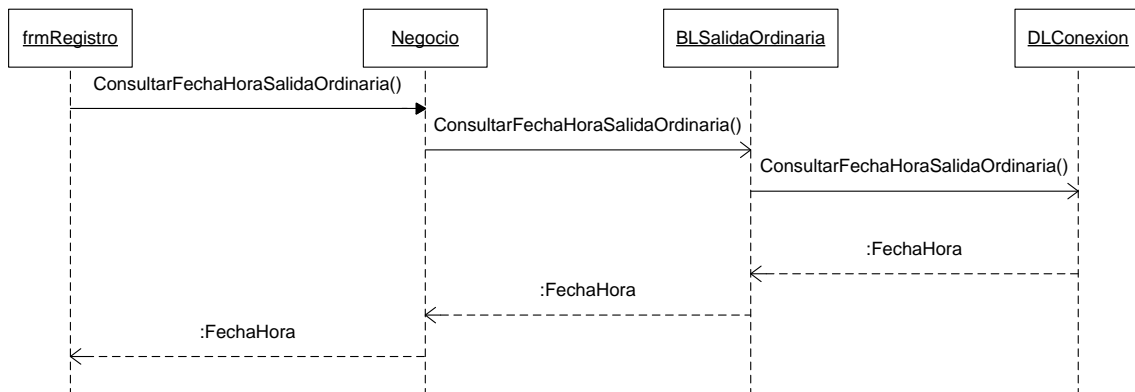
## CU-012: Consultar tipo de ausencias



## CU-013: Consultar ausencias

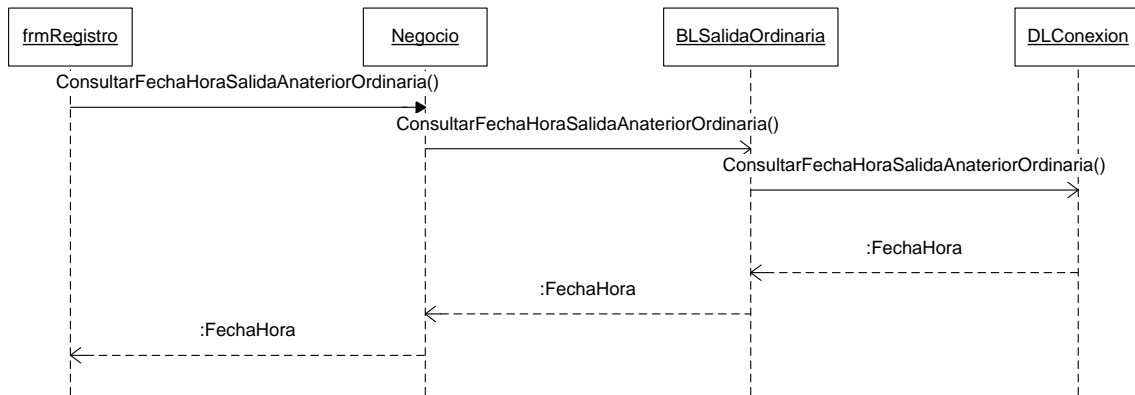


## CU-014: Consultar fecha y hora salida ordinaria

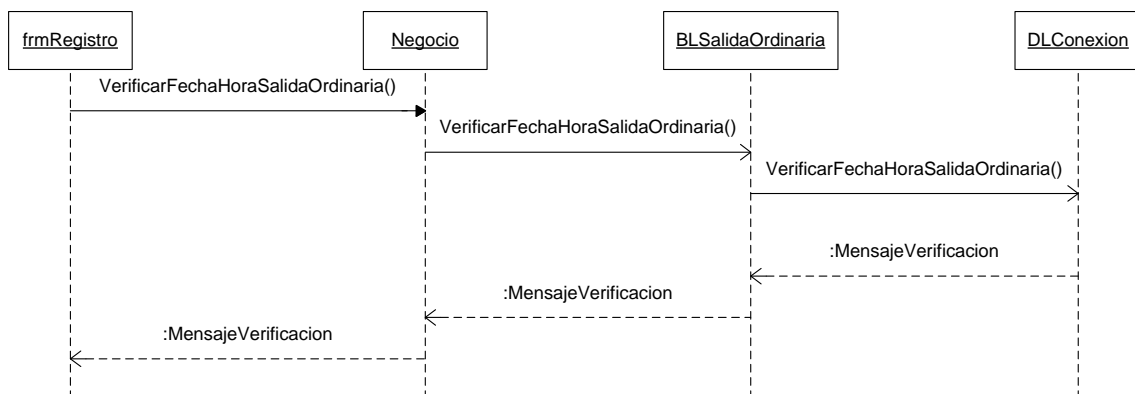




## CU-015: Consultar fecha y hora salida anterior ordinaria

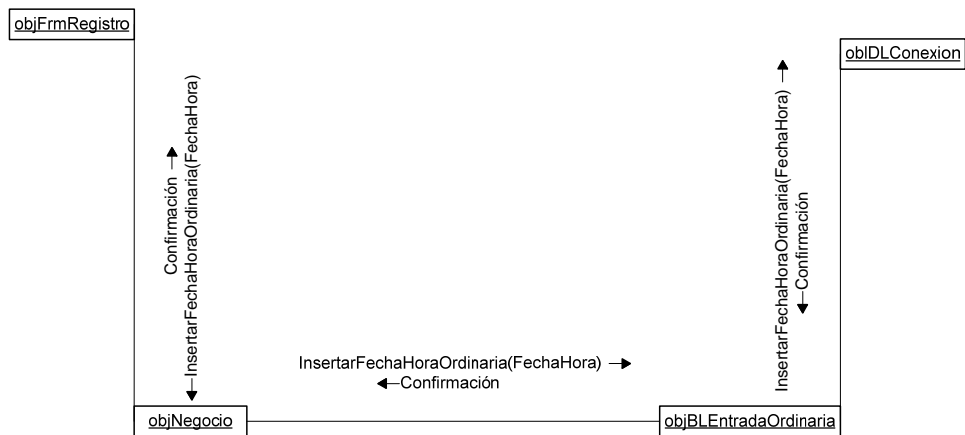


## CU-016: Verificar fecha y hora salida ordinaria

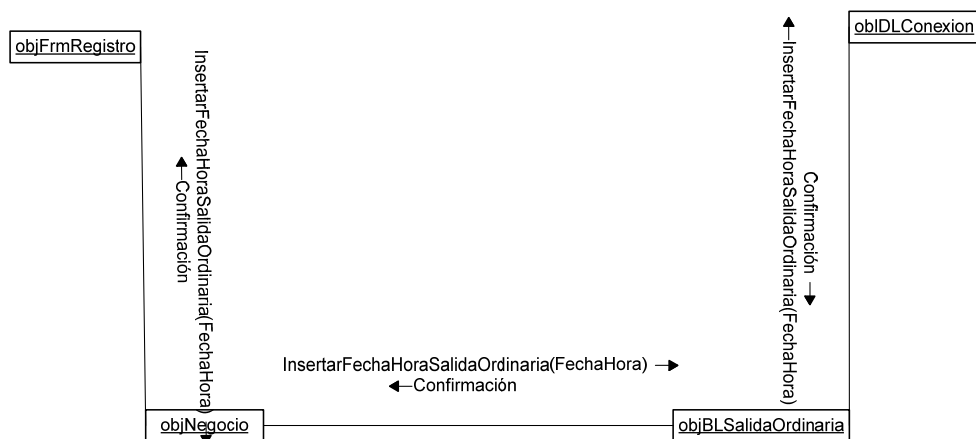


## Diagramas de Colaboración

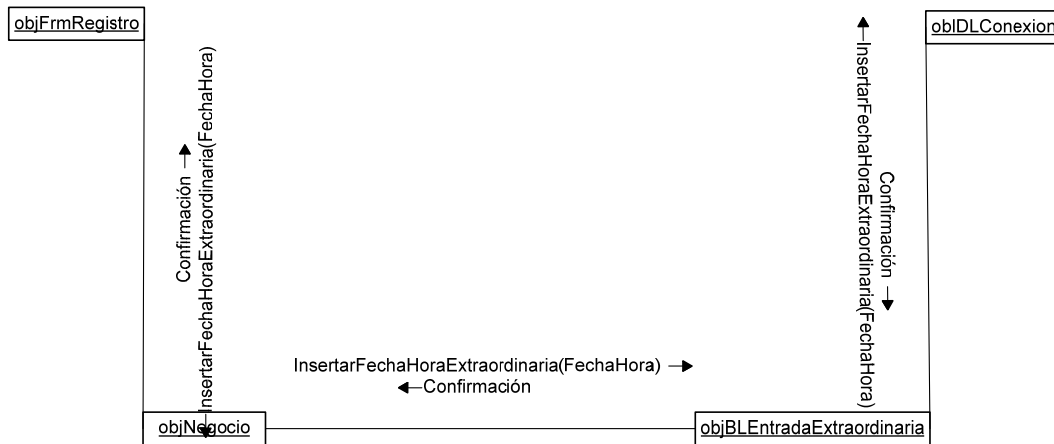
### CU-001: Insertar fecha y hora entrada ordinaria



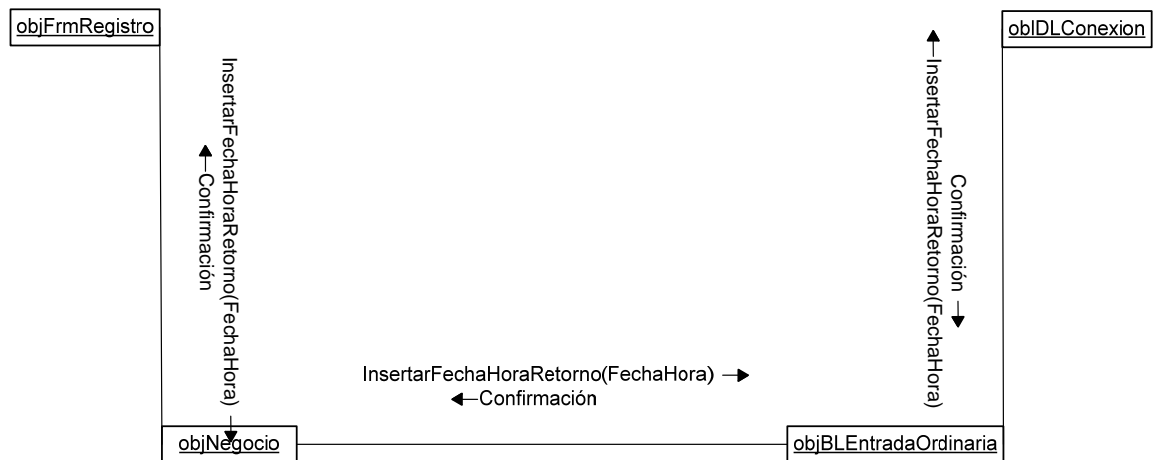
### CU-002: Insertar fecha y hora salida ordinaria



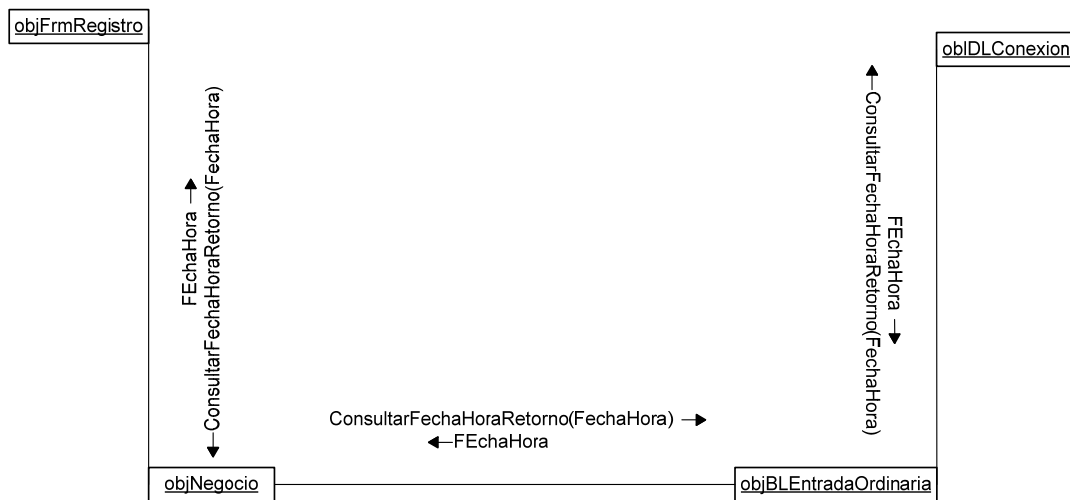
### CU-003: Insertar fecha y hora salida extraordinaria



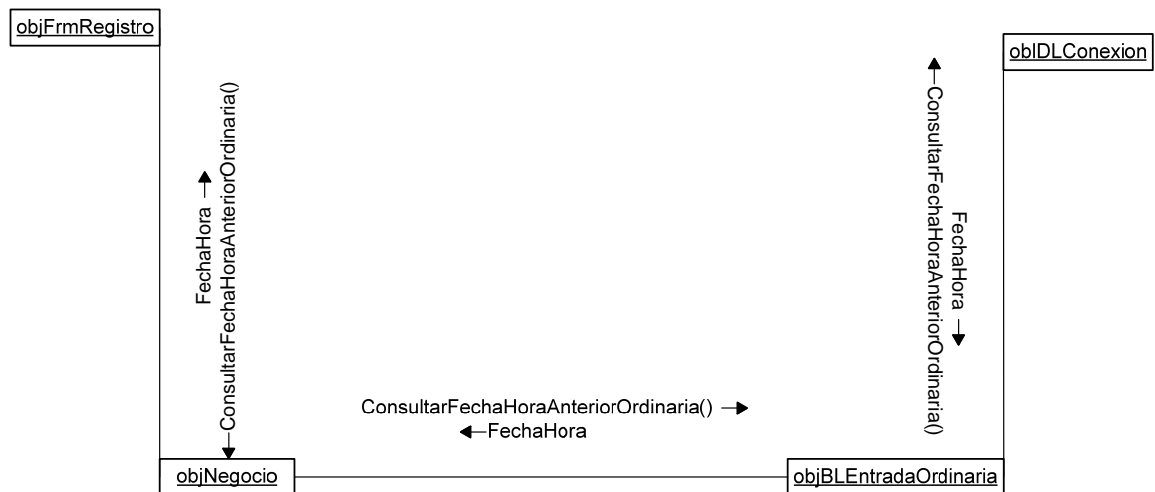
### CU-004: Insertar fecha y hora retorno



## CU-005: Consultar fecha y hora entrada ordinaria



## CU-006: Consultar fecha y hora entrada anterior ordinaria



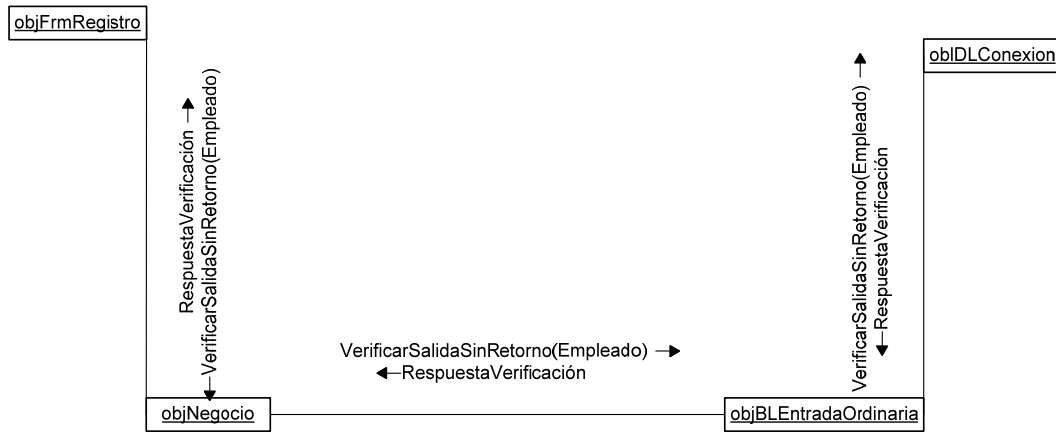
## CU-007: Verificar fecha y hora entrada ordinaria



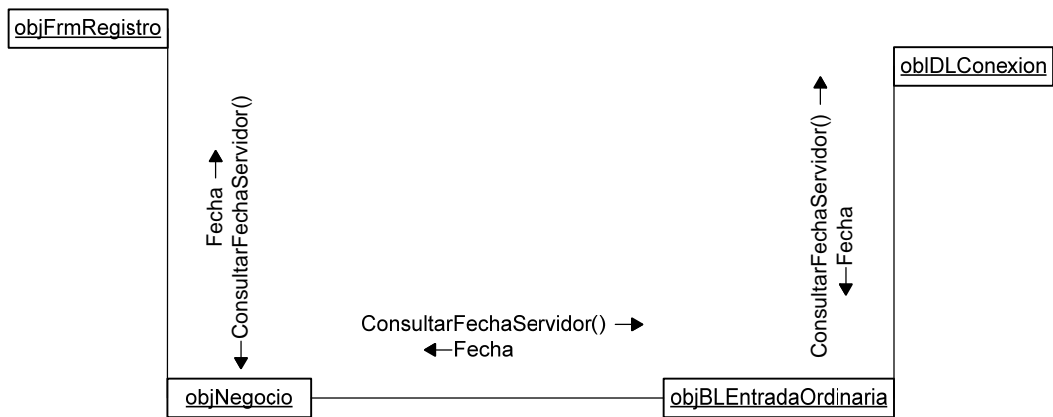
## CU-008: Consultar empleado



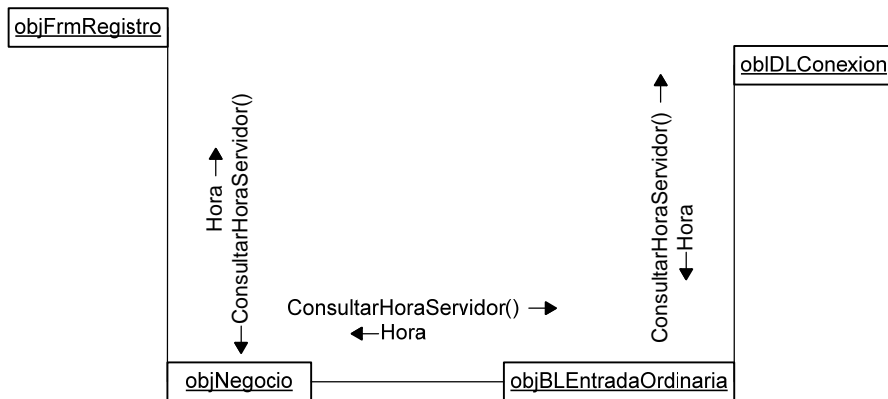
## CU-009: Verificar salida sin retorno



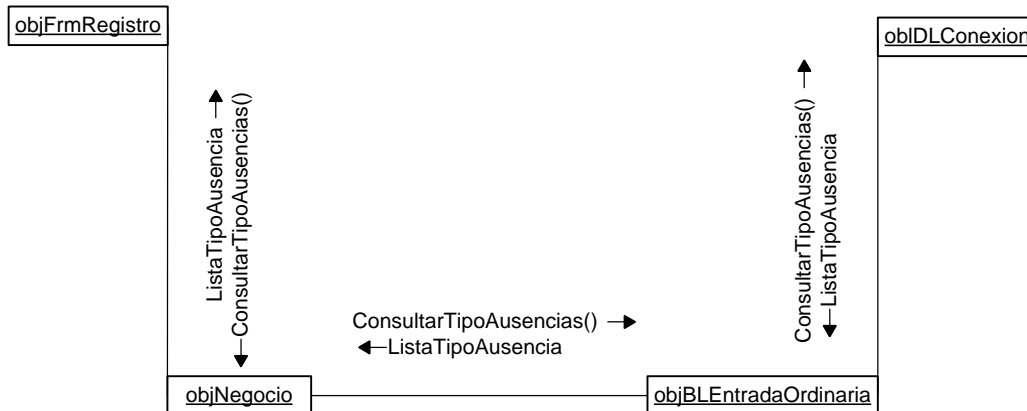
## CU-010: Consultar fecha servidor



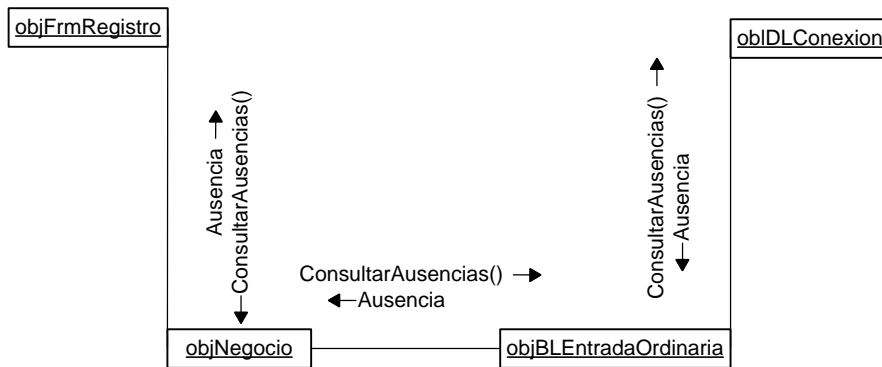
## CU-011: Consultar hora servidor



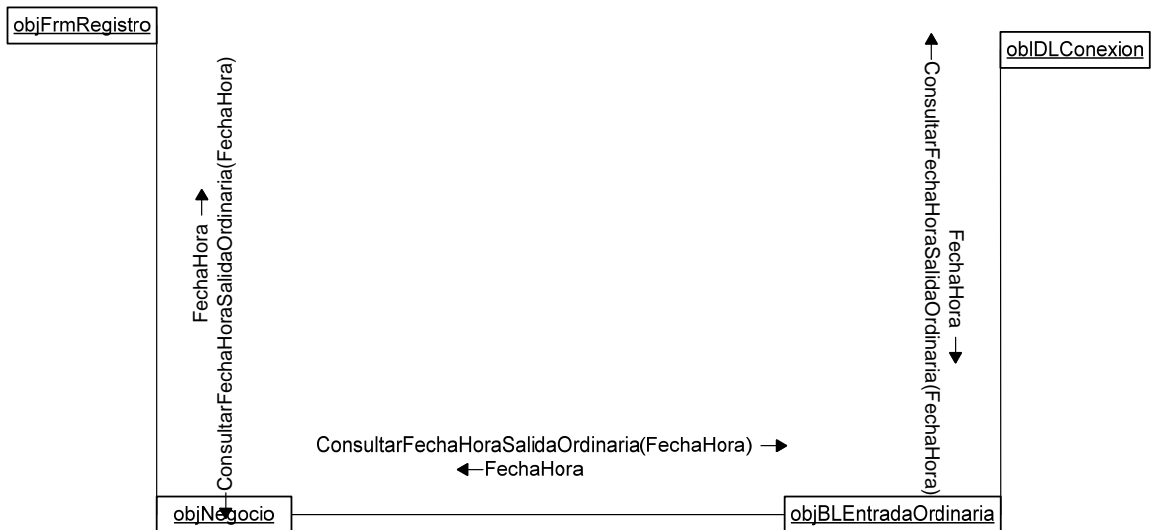
## CU-012: Consultar tipo de ausencias



### CU-013: Consultar ausencias

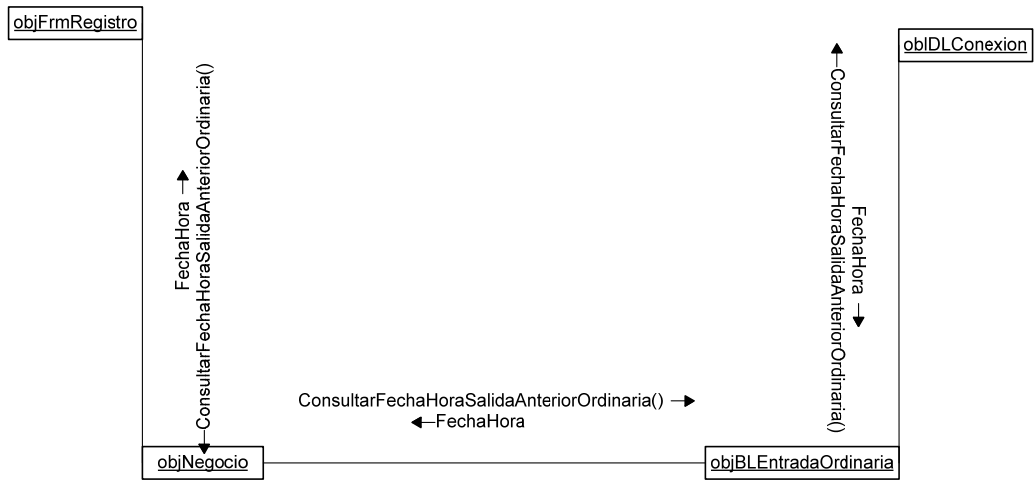


### CU-014: Consultar fecha y hora salida ordinaria





## CU-015: Consultar fecha y hora salida anterior ordinaria

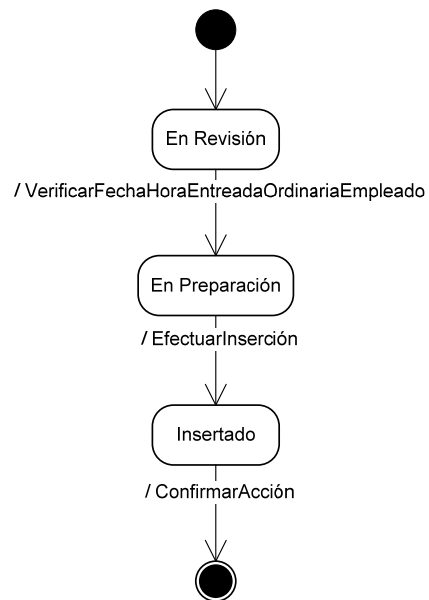


## CU-016: Verificar fecha y hora salida ordinaria

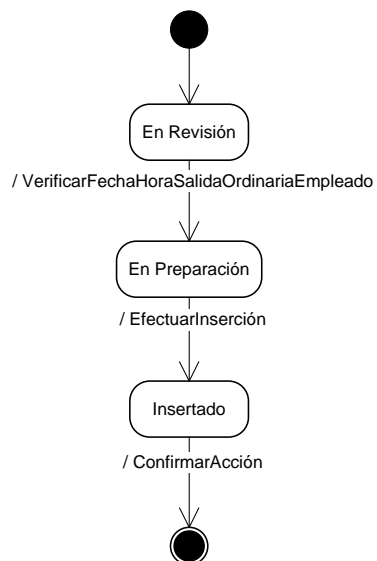


## Diagramas de Estado

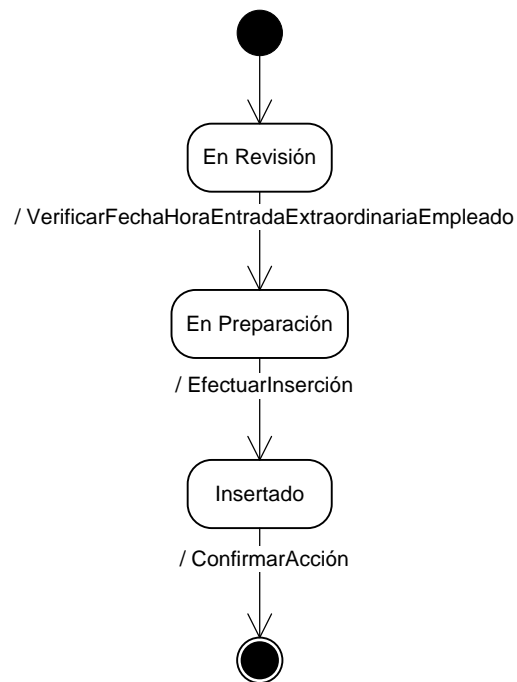
### CU-001: Insertar fecha y hora entrada ordinaria



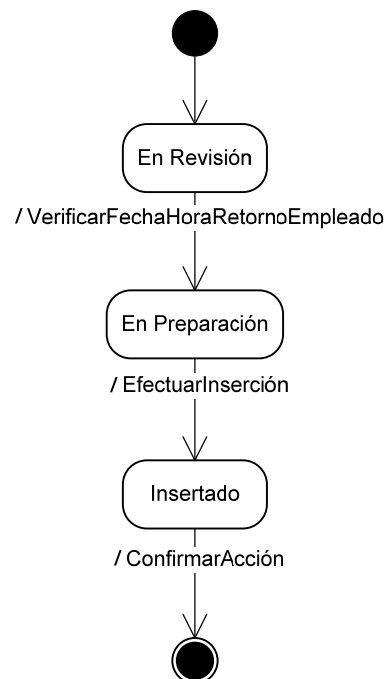
### CU-002: Insertar fecha y hora salida ordinaria



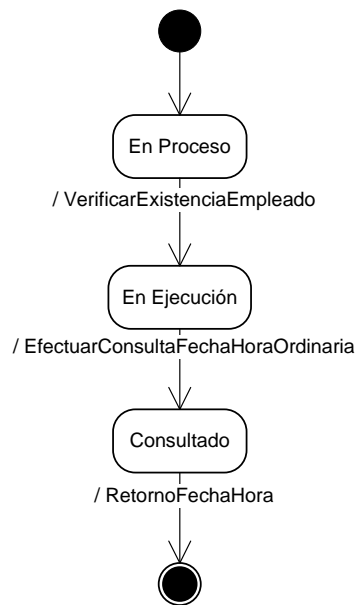
### CU-003: Insertar fecha y hora salida extraordinaria



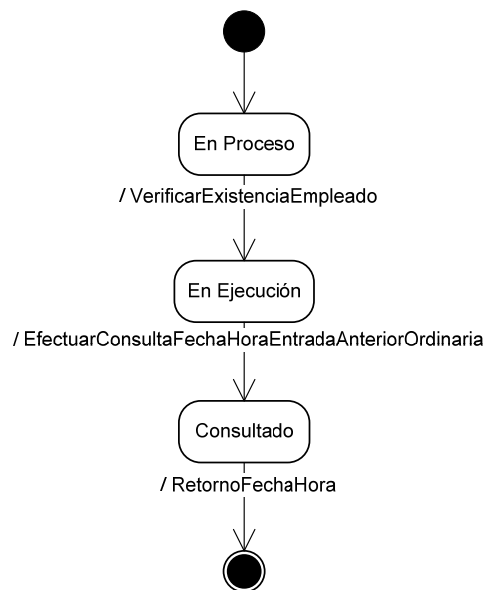
### CU-004: Insertar fecha y hora retorno



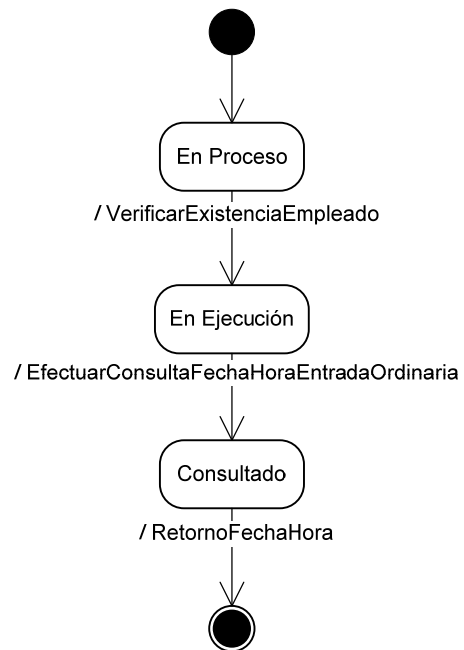
## CU-005: Consultar fecha y hora entrada ordinaria



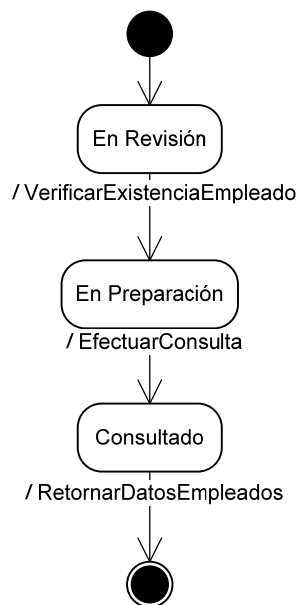
## CU-006: Consultar fecha y hora entrada anterior ordinaria



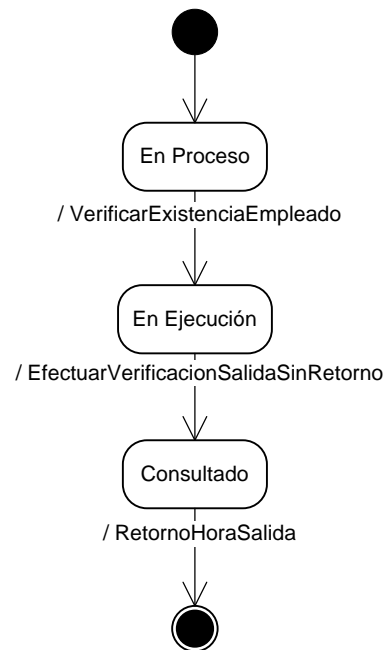
## CU-007: Verificar fecha y hora entrada ordinaria



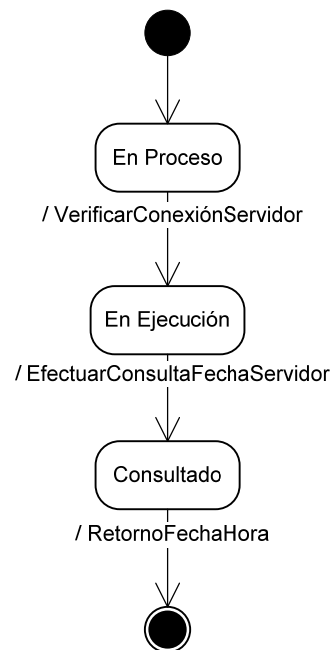
## CU-008: Consultar empleado



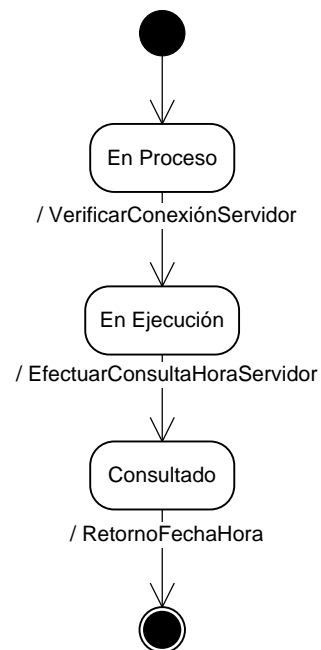
## CU-009: Verificar salida sin retorno



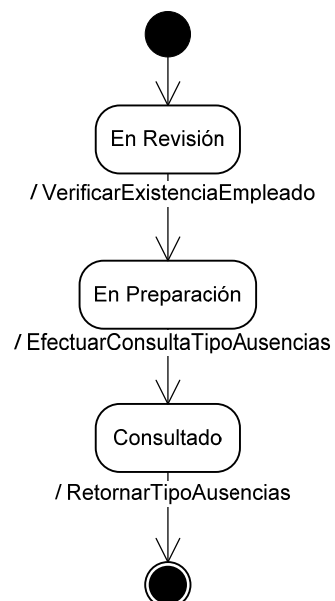
## CU-010: Consultar fecha servidor



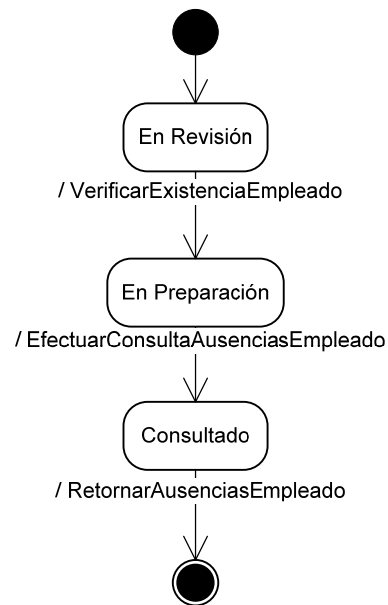
## CU-011: Consultar hora servidor



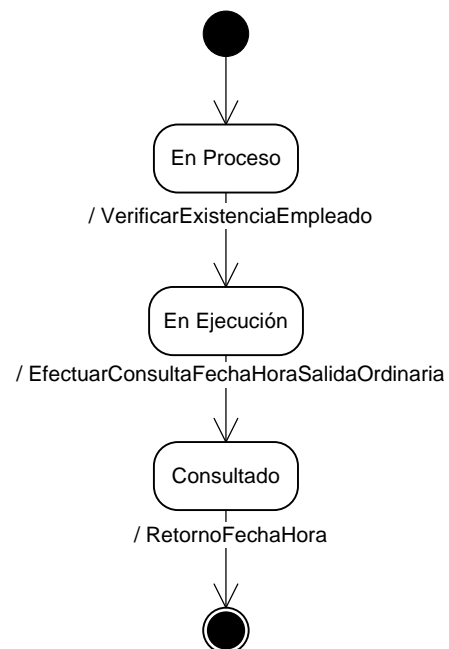
## CU-012: Consultar tipo de ausencias



## CU-013: Consultar ausencias

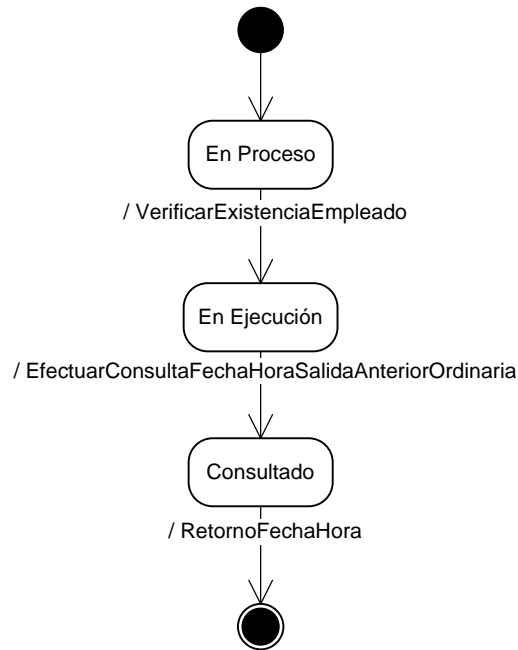


## CU-014: Consultar fecha y hora salida ordinaria

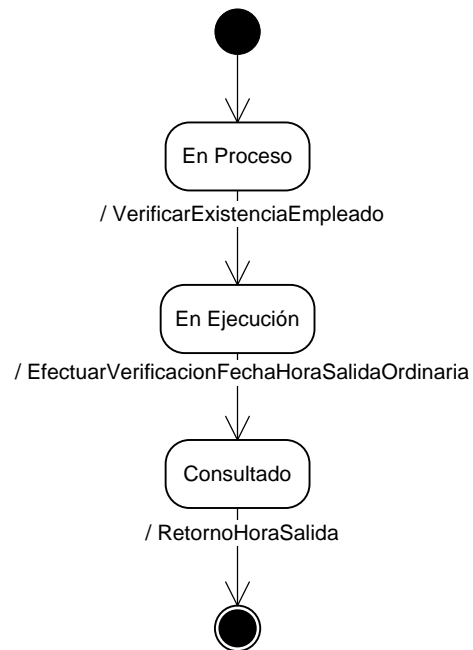




## CU-015: Consultar fecha y hora salida anterior ordinaria

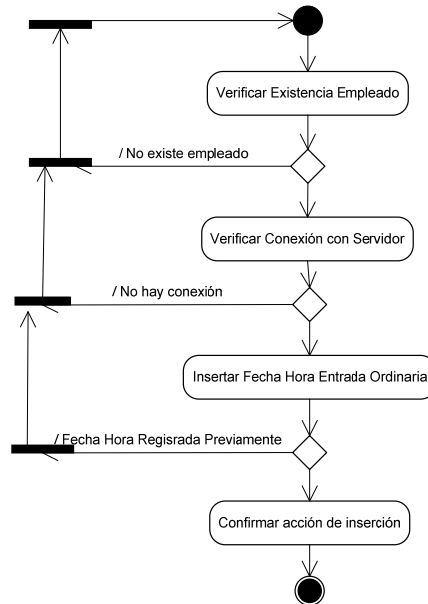


## CU-016: Verificar fecha y hora salida ordinaria

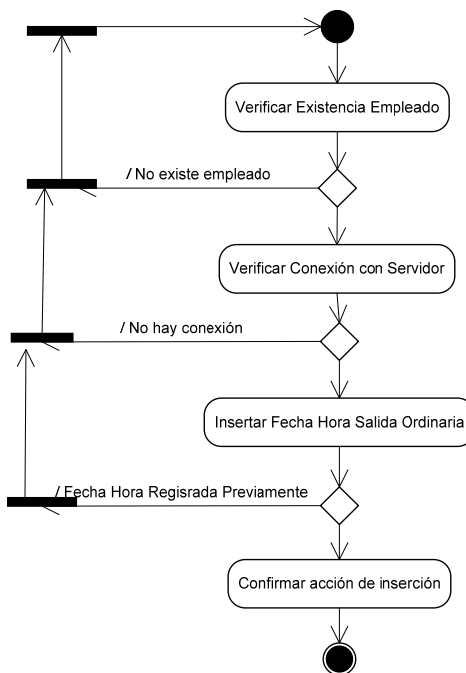


# Diagramas de Actividad

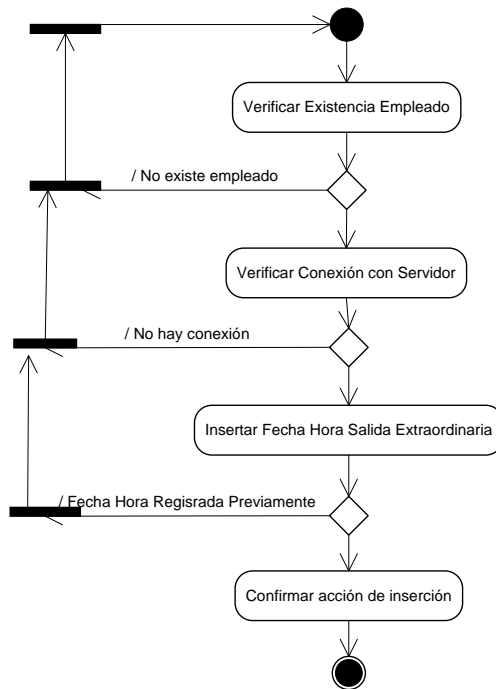
## CU-001: Insertar fecha y hora entrada ordinaria



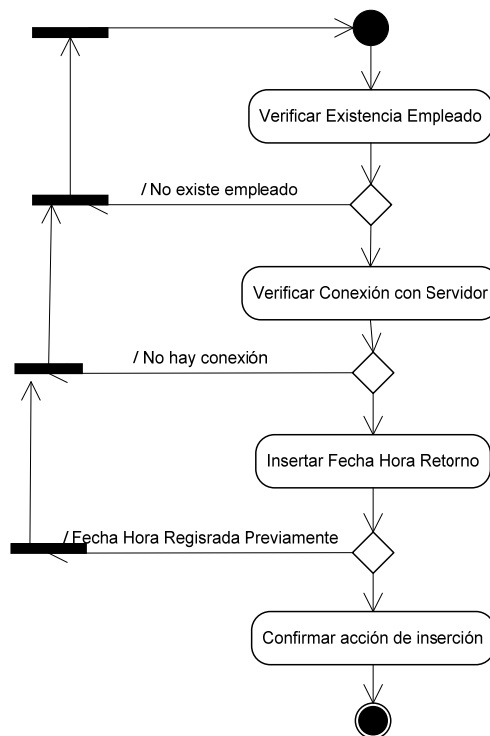
## CU-002: Insertar fecha y hora salida ordinaria



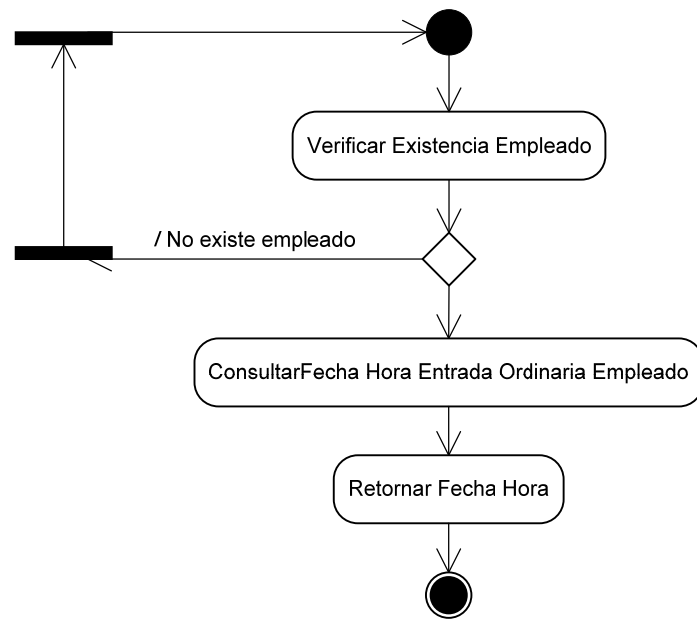
### CU-003: Insertar fecha y hora salida extraordinaria



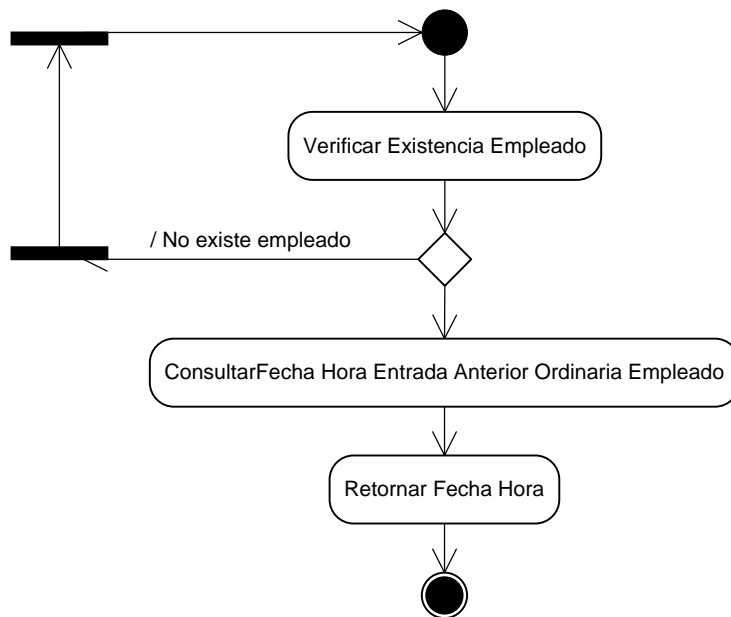
### CU-004: Insertar fecha y hora retorno



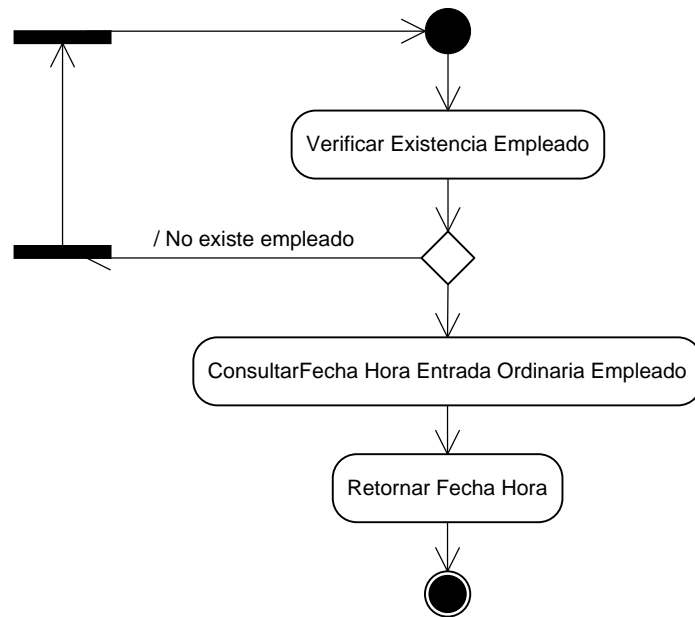
### CU-005: Consultar fecha y hora entrada ordinaria



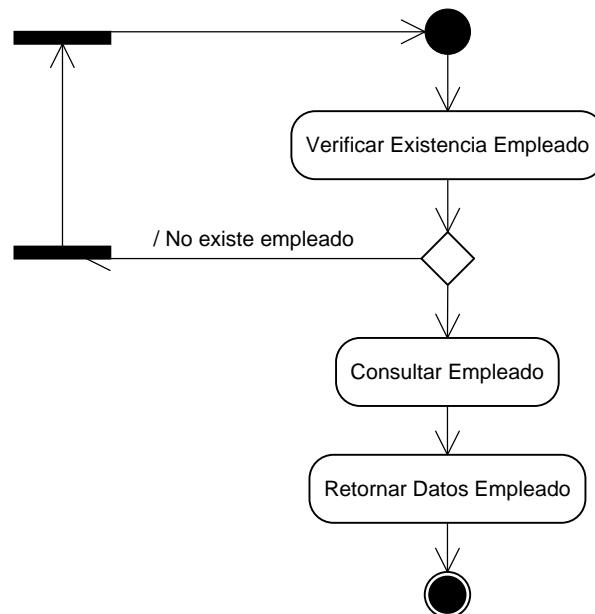
### CU-006: Consultar fecha y hora entrada anterior ordinaria



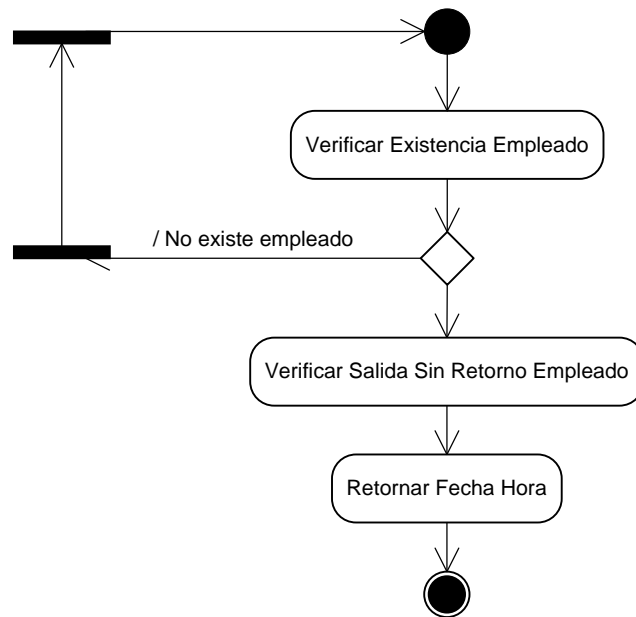
## CU-007: Verificar fecha y hora entrada ordinaria



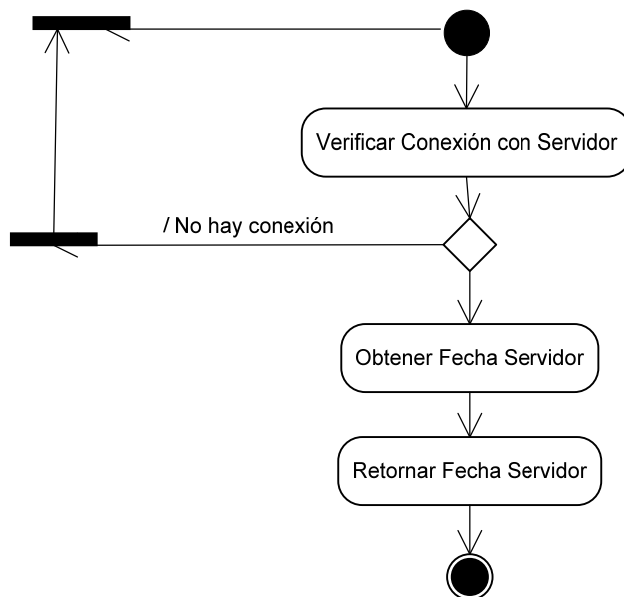
## CU-008: Consultar empleado



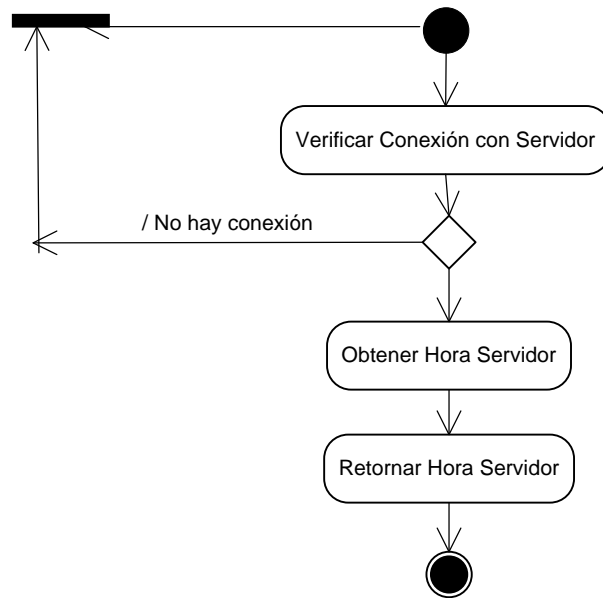
### CU-009: Verificar salida sin retorno



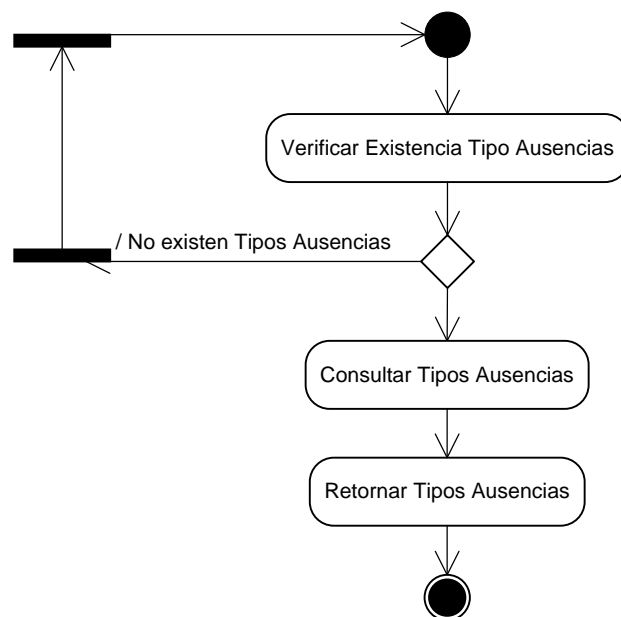
### CU-010: Consultar fecha servidor



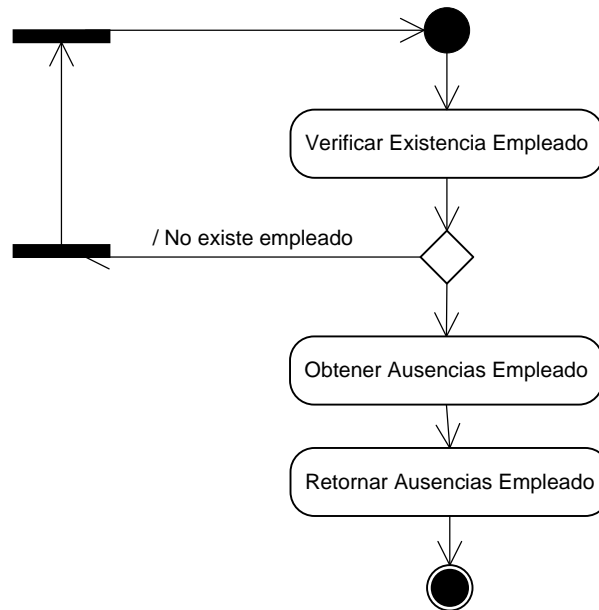
### CU-011: Consultar hora servidor



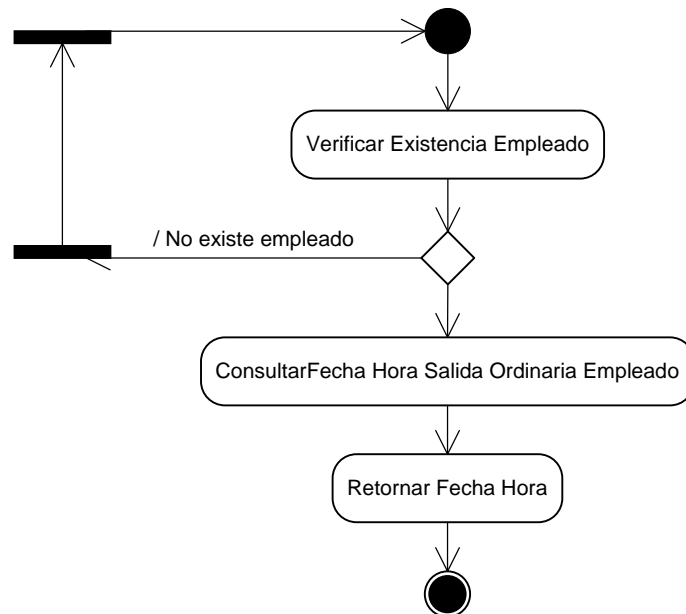
### CU-012: Consultar tipo de ausencias



### CU-013: Consultar ausencias

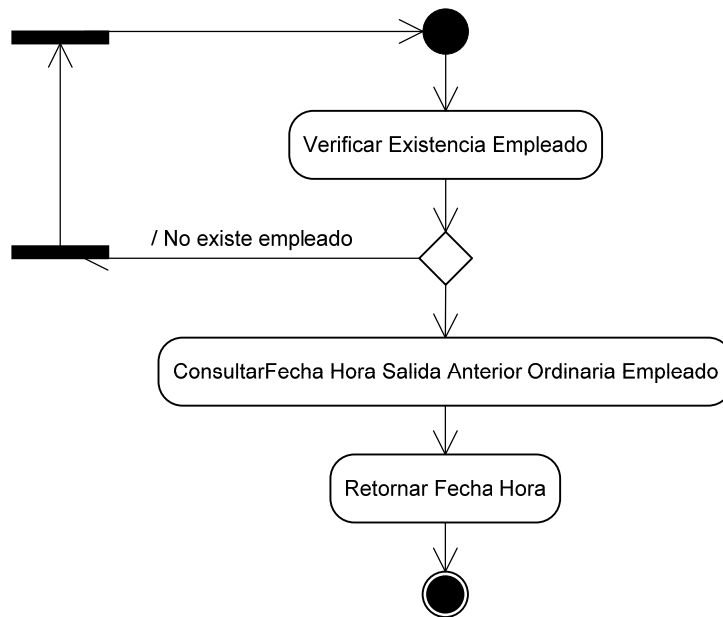


### CU-014: Consultar fecha y hora salida ordinaria

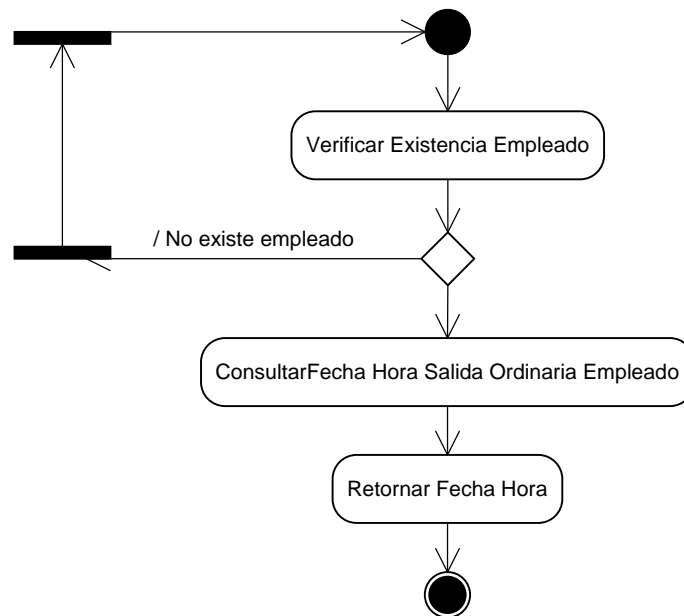




### CU-015: Consultar fecha y hora salida anterior ordinaria



### CU-016: Verificar fecha y hora salida ordinaria



## 4.3.2 CASOS DE PRUEBA



# Casos de Prueba

---

**REGASIS**

**Versión 1.0.0**

**Diana Carolina Reyes Reyes**

**2008-12-01**

## **Función de la Plantilla**

La función de esta plantilla es estandarizar y formalizar todos los aspectos que conformarán un caso de prueba.

## **Objetivos de la Plantilla**

Crear una estructura para la creación de un caso de prueba.

## **Alcance de la Plantilla**

Esta plantilla sólo se usará para elaborar los casos de prueba en el momento en que se va a crear el documento de pruebas se aplicará al proyecto.

## Revisiones de Plantilla

<b>Versión</b>	<b>Fecha de Entrega</b>	<b>Revisado por</b>
1.0.0	2007-12-21	Esteban Oviedo

## Control de versiones Plantilla

<b>Versión</b>	<b>Encargado</b>	<b>Descripción</b>
1.0.0	Kathy Guevara Meléndez	Creación de plantilla

## Revisiones

<b>Versión</b>	<b>Fecha de Entrega</b>	<b>Revisado por</b>
1.0.0	2007-09-13	Michael Bogantes

## Control de versiones

<b>Versión</b>	<b>Encargado</b>	<b>Descripción</b>
1.0.0	Diana Carolina Reyes Reyes	Creación de documento

<b>Caso de Prueba 001</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Web.Reporte.01
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Verificar el correcto funcionamiento del módulo Web de REGASIS a nivel general
<b>Requisitos asociados</b>	Caso de uso – 008
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Web de REGASIS 2- El sistema carga un listado de los empleados de Babel automáticamente. 3- Seleccionar rango de fechas (inicio y final) 4- Seleccionar los empleados a incluir en el reporte 5- Oprimir el botón “Consultar”
<b>Resultado esperado:</b>	El sistema debe mostrar el reporte de horas de entrada y salida de todos aquellos empleados seleccionados.
<b>Flujo alterno</b>	Si no se seleccionan empleados el sistema muestra un mensaje de error.
<b>Resultado alternativo esperado:</b>	El sistema despliega un mensaje indicando el error.
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.

<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	Interfaz de usuario y tiempos de respuesta.
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 002</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Web.Reporte.02
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Verificar el correcto funcionamiento del módulo Web de REGASIS en la generación del reporte
<b>Requisitos asociados</b>	Caso de uso – 008
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	<ol style="list-style-type: none"> <li>1- Ir al módulo Web de REGASIS</li> <li>2- El sistema carga un listado de los empleados de Babel automáticamente.</li> <li>3- Seleccionar rango de fechas (inicio y final)</li> <li>4- Seleccionar los empleados a incluir en el reporte</li> <li>5- Oprimir el botón “Consultar”</li> <li>6- El sistema muestra el reporte de usuarios seleccionados.</li> <li>7- Imprimir el reporte</li> </ol>
<b>Resultado esperado:</b>	El sistema debe mostrar el reporte de horas de entrada y salida de todos aquellos empleados seleccionados.
<b>Flujo alternativo</b>	Si no se seleccionan empleados el sistema muestra un mensaje de error.

<b>Resultado alternativo esperado:</b>	El sistema despliega un mensaje indicando el error.
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 003</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Web.Reporte.03
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Verificar el correcto funcionamiento del módulo Web de REGASIS en la generación del reporte en el ingreso de fechas incorrectas.
<b>Requisitos asociados</b>	Caso de uso – 008
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Web de REGASIS 2- El sistema carga un listado de los empleados de Babel automáticamente.

	<p>3- Seleccionar rango de fechas (inicio y final). Ingresar una fecha de inicio mayor a la final.</p> <p>4- Seleccionar los empleados a incluir en el reporte</p> <p>5- Oprimir el botón “Consultar”</p>
<b>Resultado esperado:</b>	El sistema debe mostrar un mensaje indicando el error en las fechas.
<b>Flujo alterno</b>	NA
<b>Resultado alternativo esperado:</b>	El sistema despliega un mensaje indicando el error.
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 004</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Web.Reporte.04
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Verificar el correcto funcionamiento del módulo Web de REGASIS en la generación del reporte en la no



	selección de empleados.
<b>Requisitos asociados</b>	Caso de uso – 008
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Web de REGASIS 2- El sistema carga un listado de los empleados de Babel automáticamente. 3- Seleccionar rango de fechas (inicio y final). 4- No Seleccionar empleados 5- Oprimir el botón “Consultar”
<b>Resultado esperado:</b>	El sistema debe mostrar un mensaje indicando un error porque no se ha seleccionado empleados.
<b>Flujo alternativo</b>	NA
<b>Resultado alternativo esperado:</b>	El sistema despliega un mensaje indicando el error.
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 005</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Win.RegistroFechaHora.01
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Verificar el correcto funcionamiento del módulo Windows de REGASIS en el proceso de registro de fechas de horas de entrada.
<b>Requisitos asociados</b>	Caso de uso – 001
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Win de REGASIS 2- Oprimir “Marcar hora de entrada”.
<b>Resultado esperado:</b>	El sistema debe registrar la hora de entrada.
<b>Flujo alterno</b>	NA
<b>Resultado alternativo esperado:</b>	NA
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 006</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Win.RegistroFechaHora.03
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Verificar el correcto funcionamiento del módulo Windows de REGASIS en el proceso de registro de fechas de horas de salida.
<b>Requisitos asociados</b>	Caso de uso – 001
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Win de REGASIS 2- Oprimir “Marcar hora de salida”.
<b>Resultado esperado:</b>	El sistema debe registrar la hora de salida.
<b>Flujo alterno</b>	NA
<b>Resultado alternativo esperado:</b>	NA
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA

<b>Corregido por:</b>	Diana Carolina Reyes Reyes
-----------------------	----------------------------

<b>Caso de Prueba 007</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Win.RegistroFechaHora.05
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Verificar el correcto funcionamiento del módulo Windows de REGASIS en el proceso de registro de fechas de horas de salida.  Ingresar una salida extraordinaria
<b>Requisitos asociados</b>	Caso de uso – 002
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Win de REGASIS 2- Oprimir “Marcar hora de salida”. (Esa hora de salida será del día siguiente)
<b>Resultado esperado:</b>	El sistema debe ingresar la salida extraordinaria.
<b>Flujo alterno</b>	NA
<b>Resultado alternativo esperado:</b>	NA
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.

<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 008</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Win.RegistroFechaHora.06
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	<p>Verificar el correcto funcionamiento del módulo Windows de REGASIS en el proceso de registro de fechas de horas de salida.</p> <p>Ingresar una ausencia.</p>
<b>Requisitos asociados</b>	Caso de uso – 013
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	<p>1- Ir al módulo Win de REGASIS</p> <p>2- Seleccionar el “Motivo de la ausencia”</p> <p>3- Ingresar la descripción</p> <p>3- Oprimir “Aceptar”.</p>
<b>Resultado esperado:</b>	<p>El sistema debe ingresar una ausencia y mostrar los siguientes datos al usuario:</p> <ul style="list-style-type: none"> <li>• Hora salida</li> <li>• Hora retorno</li> <li>• Ausencia</li> <li>• Motivo</li> </ul>
<b>Flujo alternativo</b>	NA
<b>Resultado alternativo esperado:</b>	NA

<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 009</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Win.RegistroFechaHora.07
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Verificar el correcto funcionamiento del módulo Windows de REGASIS en el proceso de registro de fechas de horas de salida.  Cancelar el ingreso de una ausencia.
<b>Requisitos asociados</b>	Caso de uso – 013
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Win de REGASIS 2- Seleccionar el “Motivo de la ausencia” 3- Ingresar la descripción

	3- Oprimir "Cancelar".
<b>Resultado esperado:</b>	El sistema debe cancelar el ingreso de una ausencia.
<b>Flujo alterno</b>	NA
<b>Resultado alternativo esperado:</b>	NA
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 010</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Win.RegistroFechaHora.08
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Verificar el correcto funcionamiento del módulo Windows de REGASIS en el proceso de registro de fechas de horas de salida.  Registrar un retorno.
<b>Requisitos asociados</b>	Caso de uso – 013

<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Win de REGASIS 2- Oprimir el botón "Registrar Retorno"
<b>Resultado esperado:</b>	El sistema debe ingresar el retorno.
<b>Flujo alternativo</b>	NA
<b>Resultado alternativo esperado:</b>	NA
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 011</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Win.RegistroFechaHora.10
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Consultar los datos del empleado
<b>Requisitos asociados</b>	Caso de uso – 008



<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Win de REGASIS 2- ingresar a la pantalla principal del sistema
<b>Resultado esperado:</b>	El sistema debe mostrar el nombre del usuario de babel y validad que este pertenezca a Active Directoy.
<b>Flujo alterno</b>	NA
<b>Resultado alternativo esperado:</b>	NA
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 012</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Win.RegistroFechaHora.11
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Consultar la fecha y hora del servidor.

<b>Requisitos asociados</b>	Caso de uso – 010 y 011
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Win de REGASIS 2- ingresar a la pantalla principal del sistema
<b>Resultado esperado:</b>	El sistema debe mostrar la fecha y hora del servidor (no de la máquina del empleado).
<b>Flujo alterno</b>	NA
<b>Resultado alternativo esperado:</b>	NA
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 013</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Win.RegistroFechaHora.12
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y</b>	Consultar motivos de las ausencias de empleados.

<b>Objetivos):</b>	
<b>Requisitos asociados</b>	Caso de uso – 010 y 011
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Win de REGASIS 2- ingresar a la pantalla principal del sistema 3- ir al módulo de ingreso de ausencias.
<b>Resultado esperado:</b>	El sistema debe mostrar un listado dentro de un comboBox que corresponde a los motivos de la ausencia.
<b>Flujo alterno</b>	NA
<b>Resultado alternativo esperado:</b>	NA
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

### Caso de Prueba 014

**Elaborado por:** Diana Carolina Reyes Reyes

<b>Código de Identificación:</b>	REGASIS.Win.RegistroFechaHora.14
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Probar la estabilidad del sistema.
<b>Requisitos asociados</b>	Caso de uso – 001
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Win de REGASIS 2- ingresar a la pantalla principal del sistema 3- desconecta la computadora del cable de red o deshabilitar la red inalámbrica. 4- Ejecutar un registro de entrada (o cualquier acción que invoque consultas a la base de datos)
<b>Resultado esperado:</b>	El sistema debe mostrar un error, pero no debe colapsar. Este debe seguir funcionando hasta que se reconecte.
<b>Flujo alternativo</b>	NA
<b>Resultado alternativo esperado:</b>	NA
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto

<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 015</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Win.RegistroFechaHora.15
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Probar la estabilidad del sistema.
<b>Requisitos asociados</b>	Caso de uso – 001
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	1- Ir al módulo Win de REGASIS 2- ingresar a la pantalla principal del sistema 3- desconecta la computadora del cable de red o deshabilitar la red inalámbrica. 4- Ejecutar un registro de entrada (o cualquier acción que invoque consultas a la base de datos) 5- Volver a establecer la conexión 6- Ejecutar un registro de entrada (o cualquier acción que invoque consultas a la base de datos)
<b>Resultado esperado:</b>	El sistema debe ejecutar la acción.
<b>Flujo alterno</b>	NA
<b>Resultado alternativo esperado:</b>	NA
<b>Evaluación de prueba</b>	
<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.

<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

<b>Caso de Prueba 016</b>	
<b>Elaborado por:</b> Diana Carolina Reyes Reyes	
<b>Código de Identificación:</b>	REGASIS.Win.RegistroFechaHora.16
<b>Nombre del Proyecto:</b>	REGASIS
<b>Descripción (Alcance y Objetivos):</b>	Probar el rendimiento del sistema.
<b>Requisitos asociados</b>	Caso de uso – 001
<b>Variables de Entrada (Inputs):</b>	NA.
<b>Flujo normal del evento</b>	<p>1- Ir al módulo Win de REGASIS</p> <p>2- ingresar a la pantalla principal del sistema</p> <p>3- Ejecutar un registro de entrada (o cualquier acción que invoque consultas a la base de datos)</p> <p>4- Verificar el tiempo de respuesta del sistema.</p>
<b>Resultado esperado:</b>	El sistema debe ejecutar la acción inmediatamente (en menos de 10 segundos).
<b>Flujo alternativo</b>	NA
<b>Resultado alternativo esperado:</b>	NA
<b>Evaluación de prueba</b>	

<b>Fecha de Ejecución:</b>	2008-12-01
<b>Ejecutado por:</b>	Diana Carolina Reyes Reyes
<b>Lugar de ejecución</b>	Oficinas centrales de Babel Software
<b>Resultados obtenidos</b>	Satisfactorio: el objeto sometido a prueba responde según el resultado esperado de la prueba.
<b>Observaciones:</b>	Deben existir registros de empleados en la Base de datos.
<b>Gravedad del error:</b>	Alto.
<b>Notas del programador</b>	
<b>Estado:</b>	Resuelto
<b>Acciones de corrección:</b>	NA
<b>Corregido por:</b>	Diana Carolina Reyes Reyes

## 4.4 CONTROL

### 4.4.1 INFORME DE RIESGOS



# Informe Riesgos #1

---

REGASIS

Versión 1.0.0

Diana Carolina Reyes Reyes

2008-09-12



## **Función de la Plantilla**

La función de esta plantilla es estandarizar y formalizar todos los aspectos dentro del informe de riesgos, el cual se vincula con los tiempos y estrategias a seguir.

## **Objetivos de la Plantilla**

1. Expresar los riesgos posibles en una etapa del proyecto.
2. Dado el caso, expresar estrategias para mitigar y reducir los riesgos.

## **Alcance de la Plantilla**

Esta plantilla se llena en las fechas marcadas en el plan de administración de para llevar el control de los riesgos del proyecto.

## Revisiones de Plantilla

<b>Versión</b>	<b>Fecha de Entrega</b>	<b>Revisado por</b>
1.0.0	2007-12-21	Esteban Oviedo

## Control de versiones Plantilla

<b>Versión</b>	<b>Encargado</b>	<b>Descripción</b>
1.0.0	Elías Conejo Cantillo	Creación de plantilla

## Revisiones

<b>Versión</b>	<b>Fecha de Entrega</b>	<b>Revisado por</b>
1.0.0	2008-09-12	Elías Conejo Cantillo

## Control de versiones

<b>Versión</b>	<b>Encargado</b>	<b>Descripción</b>
1.0.0	Diana Carolina Reyes Reyes	Creación de documento

## Tabla de contenido

<b>Función de la Plantilla .....</b>	<b>- 173 -</b>
<b>Objetivos de la Plantilla .....</b>	<b>- 173 -</b>
<b>Alcance de la Plantilla.....</b>	<b>- 173 -</b>
<b>Revisiones de Plantilla .....</b>	<b>- 174 -</b>
<b>Control de versiones Plantilla .....</b>	<b>- 174 -</b>
<b>Revisiones .....</b>	<b>- 174 -</b>
<b>Control de versiones.....</b>	<b>- 174 -</b>
<b>Tabla de contenido.....</b>	<b>- 175 -</b>
<b>Datos Importantes.....</b>	<b>- 176 -</b>

## Datos Importantes

Posición	Código del Riesgo	Prob	Impacto	Acción a seguir	Quien	Fecha
1	Atraso en el desarrollo del proyecto debido al desconocimiento del uso de las herramientas de desarrollo	75	A	<ul style="list-style-type: none"> <li>• Capacitación en .NET</li> <li>• Capacitación en Bases de datos.</li> <li>• Capacitación en AJAX</li> </ul>	Diana Carolina Reyes Reyes	2008-08-17
2	Pérdida de información debido a accesos no autorizados a las PCs.	60	M	<ul style="list-style-type: none"> <li>• Instalación de antivirus y firewall</li> </ul>	Diana Carolina Reyes Reyes	2008-08-19
3	Problemas para acceder a los recursos, servicios y configuraciones de la red	58	M	<ul style="list-style-type: none"> <li>• Actualización de componentes de Red</li> </ul>	Diana Carolina Reyes Reyes	2008-08-20
4	Accesos no autorizados	45	B	<ul style="list-style-type: none"> <li>• Restricciones al</li> </ul>	Diana Carolina	2008-08-28

	o instalación de software dañino.			acceso del servidor Arenal	Reyes Reyes	
<b>5</b>	Falta de control de usuarios y accesos a la información	38	<b>B</b>	<ul style="list-style-type: none"> <li>• Creación de perfiles de usuario en Active Directory</li> </ul>	Diana Carolina Reyes Reyes	2008-09-01
<b>6</b>	Pérdida o alteración de la información del sistema.	30	<b>B</b>	<ul style="list-style-type: none"> <li>• Respaldos constantes del software, código fuente y documentos.</li> </ul>	Diana Carolina Reyes Reyes	2008-09-10

## 4.5 CIERRE

### 4.5.1 INFORME FINAL



# Informe Final

---

**REGASIS**

**Versión 1.0.0**

**Babel Software**

**2008-12-23**

## **Función de la Plantilla**

La función de esta plantilla es estandarizar y formalizar todos los aspectos dentro de un informe final, ejemplo: los casos de uso realizados así como los casos de uso que fueron agregados en el transcurso del proyecto.

## **Objetivos de la Plantilla**

- I. Estandarizar la especificación del informe final.
- II. Crear una estructura para el informe final.

## **Alcance de la Plantilla**

Esta plantilla sólo se usará después de que termine un proyecto, después de la capacitación de los clientes y antes de la firma del documento de cierre.

## Revisiones de Plantilla

Versión	Fecha de Entrega	Revisado por
1.0.0	2008-12-23	Elías Conejo Cantillo

## Control de versiones Plantilla

Versión	Encargado	Descripción
1.0.0	Elías Conejo Cantillo	Creación de plantilla

## Revisiones

Versión	Fecha de Entrega	Revisado por
1.0.0	2008-12-23	Erick Salazar

## Control de versiones

Versión	Encargado	Descripción
1.0.0	Diana Carolina Reyes Reyes	Creación de documento

## Carta de Aprobación

Por medio de la firma de esta carta de aceptación este informe final, el cliente ratifica que los casos de uso aquí descritos fueron realizados durante el proyecto.

---

**Ing. Esteban Oviedo Blanco**

Director de Proyectos / Chief Projects Officer



## Tabla de contenido

<b>Función de la Plantilla .....</b>	<b>- 179 -</b>
<b>Objetivos de la Plantilla .....</b>	<b>- 179 -</b>
<b>Alcance de la Plantilla.....</b>	<b>- 179 -</b>
<b>Revisiones de Plantilla .....</b>	<b>- 180 -</b>
<b>Control de versiones Plantilla .....</b>	<b>- 180 -</b>
<b>Revisiones .....</b>	<b>- 180 -</b>
<b>Control de versiones.....</b>	<b>- 180 -</b>
<b>Carta de Aprobación .....</b>	<b>- 180 -</b>
<b>Tabla de contenido.....</b>	<b>- 181 -</b>
<b>Introducción .....</b>	<b>- 182 -</b>
Definiciones, Acrónimos y Abreviaturas .....	- 182 -
<b>Lista de Casos de Uso .....</b>	<b>- 183 -</b>

## **Introducción**

Este documento tiene como finalidad describir el estado de los casos de uso en el producto final, en este caso REGASIS.

## **Definiciones, Acrónimos y Abreviaturas**

- **CU:** Caso de Uso
- **REGASIS:** Sistema de registro de asistencia para empleados de Babel

## Lista de Casos de Uso

<b>Código de Caso de Uso</b>	<b>Nombre</b>	<b>Modificado</b>	<b>Completado</b>
<b>CU-001</b>	Insertar fecha y hora entrada ordinaria		X
<b>CU-002</b>	Insertar fecha y hora salida ordinaria		X
<b>CU-003</b>	Insertar fecha y hora salida extraordinaria		X
<b>CU-004</b>	Insertar fecha y hora retorno		X
<b>CU-005</b>	Consultar fecha y hora entrada ordinaria		X
<b>CU-006</b>	Consultar fecha y hora entrada anterior ordinaria		X
<b>CU-007</b>	Verificar fecha y hora entrada ordinaria		X
<b>CU-008</b>	Consultar empleado		X
<b>CU-009</b>	Verificar salida sin retorno		X
<b>CU-0010</b>	Consultar fecha servidor		X
<b>CU-0011</b>	Consultar hora servidor		X
<b>CU-0012</b>	Consultar tipo de ausencias		X
<b>CU-0013</b>	Consultar ausencias		X
<b>CU-0014</b>	Consultar fecha y hora salida ordinaria		X
<b>CU-0015</b>	Consultar fecha y hora salida anterior ordinaria		X
<b>CU-0016</b>	Verificar fecha y hora salida ordinaria		X

## V. CAPÍTULO

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 CONCLUSIONES

- Los objetivos y requerimientos planteados para el desarrollo del sistema de registro y reporte de asistencia: entradas, salidas y ausencias de los funcionarios de Babel basado en tecnología Microsoft .NET 2005 y Reporting Services, fueron alcanzados e inmersos dentro del sistema desarrollado: REGASIS.
- La aplicación de RUP y UML facilitó la ejecución del proceso de elaboración del Sistema REGASIS, permitiendo identificar claramente las necesidades de Babel y estructurar el sistema desde diferentes perspectivas tomando en cuenta a todos los involucrados en el proyecto y garantizando así el éxito del mismo.
- Durante el desarrollo del sistema REGASIS se pudo concluir que en UML los diagramas pueden ser aplicados por las empresas de acuerdo a sus necesidades, incluyendo unos y excluyendo otros dependiendo del proyecto, pero siempre tomando en cuenta que lo más importante son los casos de uso ya que estos muestran la interacción entre el usuario y el sistema.
- El sistema REGASIS es una herramienta confiable y segura que le permitirá a BABEL Software S.A. tener en tiempo real y de manera oportuna información veraz del tiempo trabajado por cada uno de sus empleados, generar reportes constantes y actualizados con información fiable en el momento que se la requiera, mientras que por otra parte sus funcionarios podrán registrar fácilmente sus horas de entrada, salida y ausencia de forma rápida y ágil desde cada una de sus PCs.

## 5.2 RECOMENDACIONES

- El sistema REGASIS, es un sistema de desarrollo a la medida, por lo que debe ser utilizado únicamente por la empresa para la que fue desarrollado, en este caso Babel Software S.A., ya que requiere del ambiente de la empresa y las configuraciones de la misma para su correcto funcionamiento y aplicabilidad.
- Los usuarios del sistema REGASIS deben ser empleados de Babel Software, poseer una computadora y pertenecer a uno de los grupos de trabajo como: Administrativos, Base de Datos, Comercial, Desarrolladores, Directores, Soporte Técnico, Administradores de proyectos, QA, Recursos humanos o Soporte de aplicaciones.
- El sistema REGASIS se podrá integrar con otros módulos ya desarrollados como el Portal Interno de Babel, el sistema de contabilidad y de recursos humanos y le será posible la implementación de mejoras y actualizaciones.
- El sistema REGASIS brinda la posibilidad de extenderle varios módulos para llevar el control de todos aquellos funcionarios que se encuentran de Outsourcing en otras empresas, previa la configuración de una VPN en las instalaciones.

## PANTALLA PRINCIPAL DEL SISTEMA REGASIS WINFORMS



PANTALLA REGISTRO DE AUSENCIAS DEL  
SISTEMA REGASIS  
WINFORMS

Registro Diario de Asistencia BABEL

**babel** software

Fecha: 18/03/2009  
Hora: 09:06:09

Bienvenido a la Aplicación Melvin Mendez

**babel** software

**Registrar Ausencia**

Motivo de la Ausencia: AsuntosBabel

Descripción:

Aceptar Cancelar

Registrar Ausencia

## PANTALLA PRINCIPAL DEL SISTEMA REGASIS WEBFORMS



Seleccione el rango de fechas:

Fecha de inicio:  

Fecha Fin:  

Seleccione los funcionarios a incluir en el reporte:

<input type="checkbox"/>	<u>Codigo</u>	<u>Nombre empleado</u>
<input type="checkbox"/>	FNC001	CRISTINA MALAVASI ANGUIZOLA
<input type="checkbox"/>	FNC002	EDGAR OVIEDO BLANCO
<input type="checkbox"/>	FNC003	ESTEBAN OVIEDO BLANCO
<input type="checkbox"/>	FNC004	FABIO FERNÁNDEZ AGUILAR
<input type="checkbox"/>	FNC005	JENNY BRENES ORTEGA
<input type="checkbox"/>	FNC006	JIMMY CHINCHILLA ARIAS
<input type="checkbox"/>	FNC007	LAURA AVENDAÑO SAGOT
<input type="checkbox"/>	FNC008	MARÍA ROSALES LÓPEZ
<input type="checkbox"/>	FNC009	MICHAEL BOGANTES RETANA
<input type="checkbox"/>	FNC010	NICKOLAS ARAYA MONTOYA

1 2 3 4 5 6 7 8 9 10 ...

Consultar



**PANTALLA RESPUESTA DE CONSULTA  
DEL SISTEMA REGASIS  
WEBFORMS**



**REPORTE DE ASISTENCIA**

<b>ASISTENCIA</b>		
<b>Día 21/07/2008 - Entrada 11:01:03 / Salida 14:06:47</b>		
<b>Motivo</b>	<b>Hora de Salida</b>	<b>Hora de Retorno</b>
<b>Día 21/07/2008 - Entrada 14:05:49 / Salida</b>		
<b>Motivo</b>	<b>Hora de Salida</b>	<b>Hora de Retorno</b>
<b>Día 21/07/2008 - Entrada 17:01:54 / Salida</b>		
<b>Motivo</b>	<b>Hora de Salida</b>	<b>Hora de Retorno</b>

## REFERENCIAS BIBLIOGRÁFICAS

El presente proyecto de investigación se ha sustentado en criterios e información que ha sido extraída básicamente de documentos a fines al tema y de páginas de internet por lo que a continuación se muestra una lista de la bibliografía y webgrafía utilizada.

- Acerca de Visual Studio  
<http://msdn.microsoft.com/es-es/vstudio/products/bb931214.aspx>
- Introducción a Visual .Net  
<http://190.40.223.217/biblioteca/Informatica/Programacion/Visual.Net/Visual%20Studio.pdf>
- Programación en Visual.NET  
<http://190.40.223.217/biblioteca/Informatica/Programacion/Visual.aspx>
- Elementos UML  
<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>
- Lenguaje Unificado de Modelado  
[http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado)
- Presentando a Reporting Services  
<http://geeks.ms/blogs/fdiaz/archive/2006/08/29/Presentando-a-Reporting-Services.aspx>
- Reporting Services  
[http://www.sqlmax.com/reportin\\_services1.asp](http://www.sqlmax.com/reportin_services1.asp)
- Tutoriales de Reporting Services  
[http://msdn.microsoft.com/es-es/library/ms170246\(SQL.90\).aspx](http://msdn.microsoft.com/es-es/library/ms170246(SQL.90).aspx)
- 2007 CENFOTEC IT, Curso de .NET Framework, FOWLER, Martin, Manual de Desarrollo Iterativo e Incremental.pdf

# **ESCUELA POLITÉCNICA DEL EJÉRCITO**

## **CARRERA DE INGENIERÍA DE SISTEMAS E INFORMATICA**

### **DECLARACIÓN DE RESPONSABILIDAD**

**Yo, Diana Carolina Reyes Reyes**

#### **DECLARO QUE:**

El proyecto de grado denominado "SISTEMA DE REGISTRO Y REPORTE DE ASISTENCIA DE BABEL DESDE CUALQUIER ESTACIÓN DE TRABAJO MEDIANTE ACCESO REMOTO, UTILIZANDO TECNOLOGÍA MICROSOFT .NET 2005 Y REPORTING SERVICES." ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mí autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, 4 de julio del 2009.

-----  
Diana Carolina Reyes Reyes

C.I. No.-1718885609

## **CERTIFICACION**

Se certifica que el presente trabajo fue desarrollado por Diana Carolina Reyes Reyes, bajo nuestra supervisión.

---

Ing. Santiago Jácome  
DIRECTOR DE PROYECTO

---

Ing. Raúl Cajas  
CODIRECTOR DE PROYECTO

---

Ing. Edison Espinoza  
COORDINADOR DE LA CARRERA DE SISTEMAS E INFORMÁTICA

---

Dr. Rodrigo Vaca  
SECRETARIO ACADEMICO