

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:
INGENIEROS DE SISTEMAS

TEMA:
IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE LA
INFORMACIÓN RELACIONADA CON EL CONTROL DE LA
ASISTENCIA DE RECURSO HUMANO DE LA EMPRESA, Y EL
REGISTRO DE ASISTENCIA PROVISTO POR UN BIOMÉTRICO CON
RECONOCIMIENTO DE IMAGEN UTILIZANDO UNA CÁMARA IP

AUTORES:
PAÚL ALEXANDER HIDALGO TOALOMBO
RAÚL ANIBAL NARVÁEZ CHICAIZA

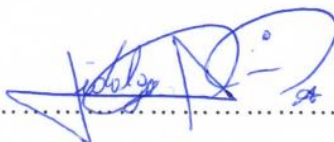
TUTOR:
ALONSO RENÉ ARÉVALO CAMPOS

Quito, septiembre de 2017

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Raúl Anibal Narváez Chicaiza y Paúl Alexander Hidalgo Toalombo, con documento de identificación N° 1722043500 y 1719481515 respectivamente, manifestamos con voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE LA INFORMACIÓN RELACIONADA CON EL CONTROL DE LA ASISTENCIA DE RECURSO HUMANO DE LA EMPRESA, Y EL REGISTRO DE ASISTENCIA PROVISTO POR UN BIOMÉTRICO CON RECONOCIMIENTO DE IMAGEN UTILIZANDO UNA CÁMARA IP, mismo que ha sido desarrollado para optar por el título de: INGENIEROS DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



PAÚL ALEXANDER HIDALGO TOALOMBO

1719481515



RAÚL ANIBAL NARVÁEZ CHICAIZA

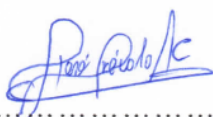
1722043500

Quito, septiembre del 2017

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico, IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE LA INFORMACIÓN RELACIONADA CON EL CONTROL DE LA ASISTENCIA DE RECURSO HUMANO DE LA EMPRESA, Y EL REGISTRO DE ASISTENCIA PROVISTO POR UN BIOMÉTRICO CON RECONOCIMIENTO DE IMAGEN UTILIZANDO UNA CÁMARA IP realizado por Raúl Anibal Narváez Chicaiza y Paúl Alexander Hidalgo Toalombo, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerados como trabajo final de titulación.

Quito, septiembre del 2017



.....
ALONSO RENÉ ARÉVALO CAMPOS
CI 1400164891

ÍNDICE GENERAL

INTRODUCCIÓN	1
OBJETIVO GENERAL	4
OBJETIVOS ESPECÍFICOS	4
CAPÍTULO 1.....	7
ESTADO DEL ARTE.....	7
1.1 Marco referencial	7
1.2 Marco teórico	7
1.2.1 Metodologías ágiles	7
1.2.2 SCRUM	8
1.2.3 BPMN.....	9
1.2.4 Diagramas de Casos de uso	9
1.2.5 Diagramas de Secuencia.....	10
1.2.6 Herramientas de desarrollo.....	10
1.2.7 Arquitectura de la aplicación.....	11
1.2.8 Mapa conceptual de la arquitectura	12
1.2.9 Arquitectura por capas	13
1.2.10 Versionamiento	16
CAPÍTULO 2.....	17
ANÁLISIS Y DISEÑO	17
2.1 Análisis	17
2.2 Requerimientos del sistema.....	17
2.2.1 Pruebas por iteración.....	25
2.2.1 Sprint Backlog.....	36
2.3 Diagrama de procesos	38
2.4 Diagramas de casos de uso	40
2.4.1 Configurar sistema	41
2.4.2 Registro y revisión de solicitudes	42
2.4.3 Tareas gerenciales	43
2.4.4 Administrador de información	45
2.5 Diagramas de secuencia	46
2.5.1 Administración de empleados	46
2.5.2 Gestión de horario	47
2.5.3 Registro y envío de solicitud	49

2.5.4 Planificación de festividades	49
2.5.5 Registro de actividades.....	50
2.5.6 Registro de asistencia	51
2.5.7 Validar solicitud	52
2.6 Modelo físico de la base de datos.....	53
CAPÍTULO 3.....	55
3.1 Estándares de programación.....	56
3.1.1 Notación.....	57
3.2 Levantamiento de la aplicación web	57
3.2.1 Base de datos.....	58
3.2.2 Manejo de la persistencia	58
3.3 Plan de pruebas	66
3.3.1 Pruebas funcionales.....	67
3.3.2 Pruebas no funcionales.....	67
3.3.3 Pruebas de rendimiento	69
3.3.3 Ambiente de pruebas	71
CONCLUSIONES	74
RECOMENDACIONES	75
LISTA DE REFERENCIAS	76

ÍNDICE DE TABLAS

Tabla 1. Plataforma tecnológica.....	11
Tabla 2. Requerimientos del sistema.....	18
Tabla 3. Resultado iteración 1	25
Tabla 4. Resultado iteración 2.....	27
Tabla 5. Resultado iteración 3.....	28
Tabla 6. Resultado iteración 4.....	32
Tabla 7. Resultado iteración 5.....	34
Tabla 8. Sprint Backlog.....	36
Tabla 9. Actores del sistema y su descripción.....	40
Tabla 10. Configurar sistema	41
Tabla 11. Registro y revisión de solicitudes	43
Tabla 12. Tareas gerenciales	44
Tabla 13. Administrador de información	45
Tabla 14. Prefijos y Sufijos	56
Tabla 15. Notación.....	57
Tabla 16. Criterios a calificar de requerimientos no funcionales.....	70
Tabla 17. Criterios de calificación	71
Tabla 18. Herramientas usadas para pruebas.....	71
Tabla 19. Plan de pruebas.	72

ÍNDICE DE FIGURAS

Figura 1 PBS de la aplicación	5
Figura 2 Definir titulo	11
Figura 3 Diagrama de secuencia de la arquitectura	13
Figura 4 Arquitectura por capas	14
Figura 5 Proceso global del sistema	39
Figura 6 Caso de uso. Configurar sistema	41
Figura 7 Caso de uso. Registro y revisión de solicitudes	42
Figura 8 Caso de uso. Tareas gerenciales	44
Figura 9 Caso de uso. Administrador de información	45
Figura 10. Diagrama de secuencia. Administración de empleados	47
Figura 11. Diagrama de secuencia. Gestión de horario	48
Figura 12. Diagrama de secuencia. Registro y envío solicitud	49
Figura 13. Diagrama de secuencia. Planificación de festividades	50
Figura 14. Diagrama de secuencia. Registro de actividades	51
Figura 15. Diagrama de secuencia. Registro de asistencia	52
Figura 16. Diagrama de secuencia. Validación de soolicitudes	53
Figura 17. Diagrama lógico de base de datos	54
Figura 18. Librería de postgresQL	58
Figura 19. Ubicación de standalone-full	59
Figura 20. Archivo de configuración standalone-full	60
Figura 21. Configuración de seguridad para autenticación	60
Figura 22. Configuración de conexión con la base de datos	61
Figura 23. Genérico para Entity Manager	62
Figura 24. Entity SegUsuario	63

RESUMEN

Las empresas y microempresas poseen varias responsabilidades, una parte fundamental de éstas son sus trabajadores, a los cuales se les asigna un horario de trabajo y está encargado en la mayoría de los casos de ser controlado por un departamento de recursos humanos. Generalmente el control de asistencia se lo realiza manualmente, pues no cuentan con una aplicación que les permita gestionar este proceso. Aunque existen empresas que cuentan con varios tipos de biométricos, los aplicativos que lo manejan son muy limitados y están orientados solamente a la descarga de información de los registros de sus funcionarios. El presente proyecto plantea un sistema de gestión de la información relacionada con el control de asistencia de recurso humano de la empresa que tiene como objetivo solventar este problema y facilitará la gestión del registro de asistencia del personal, permisos, entre otras actividades que tiene a su cargo el departamento de recursos humanos, basándose en las disposiciones del código de trabajo provisto por el ministerio de relaciones laborales. El objetivo de este proyecto es brindar una información clara, oportuna y objetiva, acorde a lo que los entes de control dispongan, ahorrando recursos y tiempo.

ABSTRACT

Enterprises and micro-enterprises have several responsibilities, a fundamental part of these are employees, he is assigned to which working hours and who is responsible in the majority of cases of control by a human resources department. Usually control of assistance it is done manually, because they do not have an application that allows them to manage this process. Although there are companies that have various types of biometric applications that manage it are very limited and are geared only to the download of information from the records of its officials. The present project proposes a management system of information related to the control of assistance of human resource of the company that aims to solve this problem and facilitate management of personnel attendance record, permits, among other activities is in charge the Department of human resources, based on the provisions of the code of work provided by the Ministry of Labor Relations. The objective of this project is to provide clear, timely information and objective, according to what the control bodies available, saving time and resources.

INTRODUCCIÓN

El presente proyecto se aplica al departamento de recursos humanos de una empresa, pues entre las diferentes funciones que deben cumplir se encuentra la de gestionar el recurso humano, es decir, horarios laborales, permisos, vacaciones, atrasos, horas extra, entre otras funciones. La mayoría de empresas lleva este control a partir de un biométrico, pero los reportes generados son limitados, es por esto que este proyecto muestra las ventajas de utilizar este aplicativo como herramienta de apoyo para gestionar el personal de una empresa.

Las empresas tienen que registrarse bajo el del código de trabajo que se encuentre vigente en el registro oficial, en la que abarca varios derechos y obligaciones que deben cumplir con los empleados, por consiguiente, el presente proyecto ofrece opciones de parametrización que pueden estar sujeto a cambios según se reforme el código de trabajo, o a su vez mediante decreto se disponga alguna indicación con lo que respecta a horarios laborales o días festivos, entre otros parámetros.

Las empresas en la actualidad manejan este proceso por medio de hojas de cálculo a partir de reportes limitados a horas de ingreso y salida del personal, es decir, hasta cierto punto es un proceso manual, lo que causa una posible falla en la digitación de valores; no se tiene actualizado periódicamente los datos del área del departamento de recursos humanos; es por esto que se decidió realizar un software que facilite la gestión de la información relacionada con el control de la asistencia de recurso humano de la empresa, y el registro de asistencia provisto por un biométrico.

El sistema informático se construyó utilizando herramientas como: Eclipse, para la codificación, ya que es compatible con todas las especificaciones de Java EE7; JSF 2.0 y para la arquitectura de desarrollo web; PostgreSQL, como motor de base de

datos relacional y Primefaces-Sentinel, para facilitar el desarrollo de la capa de presentación de la aplicación.

Desde los inicios del siglo actual los departamentos de recursos humanos desempeñan actividades dentro de la empresa trascendentales para el desarrollo y crecimiento de la misma y que, con el avance del tiempo se les ha ido atribuyendo varias funciones, una de las más importantes son las jornadas laborales establecidas por el código de trabajo. Por este motivo conforme pasa el tiempo se ha ido encontrando varias formas de controlar lo que son los horarios laborales dentro de la empresa.

Una de las herramientas más usadas hasta hace poco en la mayoría de las empresas ha sido un reloj para el control por impresión de tarjeta llamado “reloj electromecánico” que conforme ha ido avanzando la tecnología fueron quedando obsoletos.

Los Relojes electromecánicos no poseen ningún tipo de conexión a PC por lo que no es posible realizar procesamiento posterior de todas las fichadas almacenadas empleando software especializado. El medio de identificación utilizado es el de tarjeta de cartulina. Cada uno de los registros que se realicen va siendo almacenado mediante la impresión de la hora en el sector correspondiente al día en curso, en una tarjeta de cartulina especialmente diseñada.

Para determinar el total de horas trabajadas por cada empleado debe realizarse un relevamiento de cada tarjeta. (Sistemas, 2013).

Entre las desventajas que poseía esta herramienta es que las tarjetas para registrar horas de ingreso y salida eran unipersonales, y al ser un documento físico se podría correr el riesgo de que se extravíen. Para luego pasar manualmente a una hoja de cálculo y con el riesgo de que exista errores de tipeo.

Conforme ha ido avanzando la tecnología han aparecido diferentes tipos de biométricos electrónicos que han tenido gran acogida en las empresas, pero pese a este avance los reportes y configuraciones que lo controlan son limitados y en muchos de los casos terminan pasando y calculando las horas trabajadas, y demás parámetros en una hoja de cálculo.

El propósito de este producto tecnológico es facilitar la gestión del registro de asistencia del personal, permisos, asignación de vacaciones, cálculo de horas extra del empleado, creación y configuración de horarios de trabajo, calendarización de fechas festivas, generación de reportes, etc. basándose en las disposiciones del código de trabajo provisto por el ministerio de relaciones laborales.

Se ha visto la necesidad de implementar el presente proyecto para dar un beneficio de ahorro de tiempo y recursos al departamento de recursos humanos, pues como tomando como base experiencias laborales de personas dentro de empresas tanto públicas como privadas, al poseer sistemas biométricos de todo tipo no han logrado satisfacer sus necesidades acordes a sus expectativas. Ejemplificando este proceso, se ha logrado tener conocimiento de que a sus empleados cada quincena o su vez mensualmente presenten informes de asistencia pidiendo el informe a un departamento de nómina, para luego de haberlo pasado a una plantilla de Excel predeterminada según la necesidad de cada empresa ir al departamento de talento humano a entregar dicho informe.

OBJETIVO GENERAL

Implementar un sistema de gestión de la información relacionada con el control de la asistencia de recurso humano de la empresa, y el registro de asistencia provisto por un Biométrico con reconocimiento de imagen utilizando una cámara IP.

OBJETIVOS ESPECÍFICOS

Diseñar un módulo de configuración de horarios de trabajo y tipificación de horas extras.

Diseñar un módulo de configuración de días festivos y laborables de acuerdo a las necesidades de la empresa.

Diseñar un módulo de registro del personal

Diseñar un módulo de seguridad (acceso al sistema de acuerdo a perfiles)

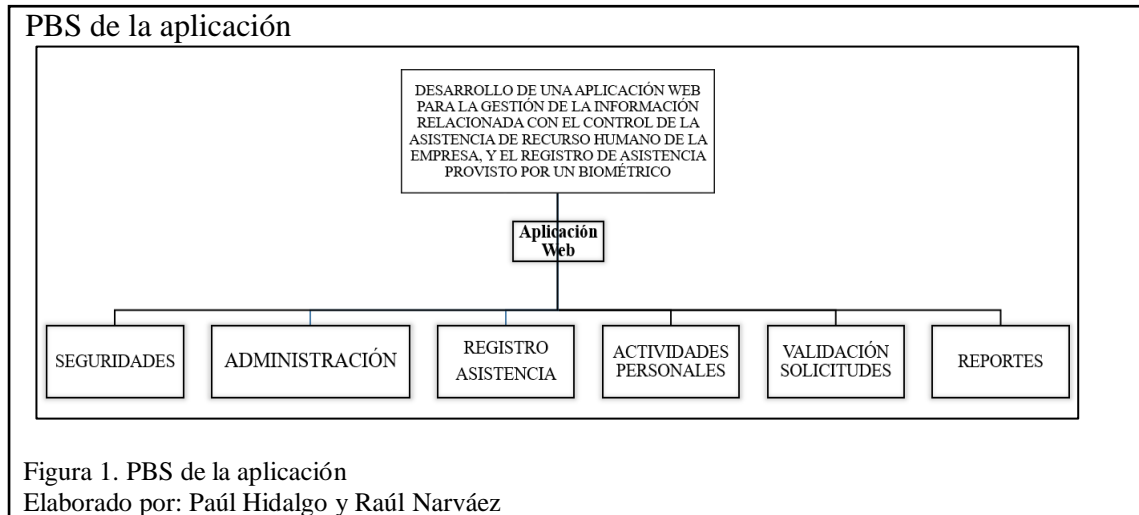
Diseñar un módulo para el cálculo de las horas laboradas por el funcionario y sus horas extras.

Diseñar un módulo para el registro de permisos.

Diseñar un módulo de registro de periodos de vacaciones.

Diseñar un módulo de reportes parametrizables con la información del registro de asistencia, control de atrasos, planificación de vacaciones anuales, días festivos, días laborables y horas extras por periodo, y funcionario.

El alcance del producto se especifica mediante el diagrama de estructura de desglose del producto o product breakdown structure (PBS).



Según la Figura 1 el presente aplicativo va abordar seis módulos en general que son: el de seguridades, en cargado manejar los perfiles y usuarios de la aplicación; administración, la cual realiza como base principal la gestión de horarios laborales; registro asistencia, ya que no siempre se realiza la asistencia mediante un biométrico si no que en ocasiones necesita un registro manual; actividades personales, especialmente usado para empleados que hayan realizado horas extra y puedan justificar; validar solicitudes, debido a la jerarquía que se tiene en una empresa se tiene que validar por un ente superior; y finalmente, los reportes, que pueden ser: personales, administrativos que constan de solicitudes y tiempos laborales.

El marco metodológico se encuentra basado en cuatro puntos fundamentales:

El estado del arte, que trata de la definición de una manera más simple los requerimientos, además de delimitarlos sin perder la perspectiva del tema planteado; situación actual, que necesariamente deben estar alineados con el código de trabajo que es el que directamente regula el proceso; diseño y desarrollo, se usó SCRUM para

la gestión del desarrollo del software y UWE que permite generar la documentación técnica del proceso de desarrollo de software; finalmente la validación del producto, que es un proceso que recae directamente en las pruebas correspondientes a cada iteración y a las pruebas de integración una vez terminado el producto. Además, que el objetivo primordial es el de satisfacer los requisitos del usuario y que se encuentre con la calidad esperada.

CAPÍTULO 1

ESTADO DEL ARTE

1.1 Marco referencial

De acuerdo con (tecnologias-informacion, 2016) al abordar el tema de sistemas de gestión de recursos humanos y al ejercer un papel vital en garantizar el buen funcionamiento de una empresa, menciona que ha sido una área que no ha pasado inadvertida por la industria del software. Entre las múltiples aplicaciones que existen en el mercado existen módulos que se dedican exclusivamente a cubrir la gestión de recursos humanos de personal y que sea atendido eficazmente el fondo de capital humano.

Sin embargo, la empresa de Seguridad y Equipos de Alta Tecnología (2016) pese a tener sistemas biométricos de diferente tipo, tienen una limitación al momento de generar sus reportes, pues cuenta en la mayoría de equipos solo con reportes de entrada y salida y no tiene en cuenta algunos parámetros que rigen a la empresa según el código de trabajo.

1.2 Marco teórico

1.2.1 Metodologías ágiles

Se ha creado las metodologías ágiles por lo cambiante que se ha vuelto nuestro ambiente de desempeño, por este motivo las metodologías tradicionales han ido quedando obsoletas pues no se logra cumplir con las expectativas de los beneficiarios, o a su vez se dificulta realizar algún cambio por la naturaleza de la metodología tradicional.

Existe una organización sin fines de lucro denominada *The Agile Alliance* que fue creada en febrero del 2001 tras una reunión de expertos en la industria del software la

cual se “dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida es el Manifiesto Ágil, un documento que resume la filosofía “ágil”. ” (Torres, 2007).

1.2.2 SCRUM

Entre todas las metodologías ágiles que existen se ha escogido SCRUM por los riesgos de cambios que existen durante el proceso, además se ejecuta en fases cortas de dos a cuatro semanas, es flexible a los cambios a mitad del proyecto, además que es muy funcional con equipos pequeños, esto no quiere decir que en proyectos grandes no funcione, pero no es muy recomendable hacerlo en estos casos.

Entre los aspectos básicos que se debe tener en cuenta son los artefactos, entre ellos tenemos el Product Backlog correspondiente a los requerimientos del usuario que a su vez se va priorizando según el criterio del Product Owner (Propietario del producto) y la Historia de usuario correspondiente a un documento funcional conformado por una sentencia y criterios de aceptación, que son parte de los insumos. Cabe recalcar que estos insumos son dinámicos.

El Project manager o un gestor de proyecto maneja la comunicación a cargo del propietario del producto, requerimientos de alcance, planeación y control de tiempo, controles presupuestarios y temas de equipo.

Los roles principales del negocio de SCRUM son el Product Owner, maestro SCRUM y los desarrolladores, luego pero no menos importante se encuentran las personas involucradas en el negocio o partes interesadas.

Para obtener la información necesaria por parte de las partes interesadas se lo realiza de diferentes maneras y existen varias técnicas y herramientas, una de las más comunes y usada es la entrevista, pero también existen cuestionarios, documentación

relacionada con el proyecto que puede ser pública y que a su vez se tiene un fácil acceso además de videos o grabaciones, etc.

1.2.3 BPMN

BPMN es una herramienta que tiene una vinculación directamente con el negocio, además que es de gran utilidad para modelar dichos procesos de negocio de una forma estándar, en breve han sido descritas las principales funcionalidades del Business Process Model and Notation (BPMN), que es español el Modelo y Notación de Procesos de Negocio.

El principal objetivo de BPMN es proveer una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio. Entre ellos están los analistas de negocio técnicos, quienes definen los procesos; los desarrolladores técnicos, que son los responsables de implementar los procesos y los gerentes o administradores del negocio, cuya responsabilidad es monitorear y gestionar los procesos (Gómez Estupiñán, 2014, pág. 15).

Cabe resaltar que BPMN describe una notación que no es difícil para realizar los flujos y lo mejor de esto es que no tiene una vinculación con el entorno de implementación que se ocupe.

1.2.4 Diagramas de Casos de uso

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso) (Salinas Caro & Hitschfeld Kahler, s.f.).

Este artefacto no necesita que sea tan detallado, si no que se resumen a los casos de uso con una visión general de lo que se quiere plasmar.

1.2.5 Diagramas de Secuencia

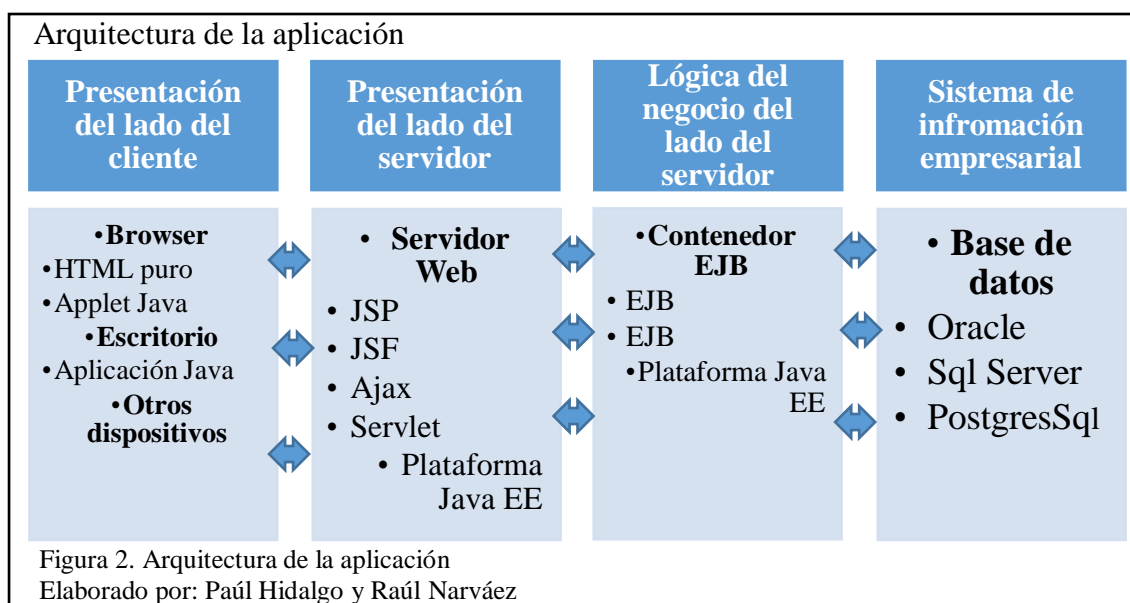
Los diagramas de secuencia se pueden modelar a partir de los casos de uso ya que al tener identificados los actores y las actividades que realiza, en este diagrama se plasma el intercambio de los mensajes, entradas y salidas entre los elementos de nuestro sistema. Se encuentra dentro de la vista de interacción pues proporciona una vista integral del comportamiento de un sistema.

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones. Los diagramas de secuencia no están pensados para mostrar lógicas de procedimientos complejos. (Systems, 2017).

1.2.6 Herramientas de desarrollo

Como plataforma para el desarrollo de aplicaciones se usa Java Enterprise Edition (JEE) para el caso la edición 7. En lo que respecta a la capa de aplicación se pueden utilizar varios modelos de componentes, según se necesite, para este caso se ha utilizado Enterprise Java Beans 3.1 y en la capa de presentación se está usando el modelo de componentes JavaServer Faces 2.0.

El modelo conceptual de la capa de aplicación se los describe a continuación:



Las herramientas de desarrollo a ocupar maximizan la productividad de los desarrolladores, permiten el uso de una instancia del servidor de aplicaciones dentro del entorno de desarrollo, mencionado esto se ha usado como herramienta de desarrollo a Eclipse que es un entorno de desarrollo de código abierto.

Plataforma tecnológica

La plataforma tecnológica para la muestra de la aplicación está constituida de la siguiente manera:

Tabla 1. Plataforma tecnológica

Sistema operativo	Windows 10
Servidor de aplicaciones	JBoss Enterprise Application Platform 8
Gestor de base de datos	Postgres 9.5
Entorno de desarrollo	Eclipse

Nota: Plataforma tecnológica de la aplicación
Elaborador por: Paúl Hidalgo y Raúl Narváez

1.2.7 Arquitectura de la aplicación

La arquitectura planteada para esta aplicación consta con un servidor de aplicaciones que está constituido de un contenedor Web y contenedor de componentes de negocio. Este servidor contendrá toda la aplicación formando lo que se conoce como capa

media, mientras que es en la capa física donde reside toda la lógica del negocio y los clientes son browsers o aplicaciones standalone, que acceden a esta capa a través del contenedor web.

El contenedor web es el responsable de construir la capa de presentación que será entregada a los clientes y ejecutar acciones del negocio utilizando los componentes que ya existen.

1.2.8 Mapa conceptual de la arquitectura

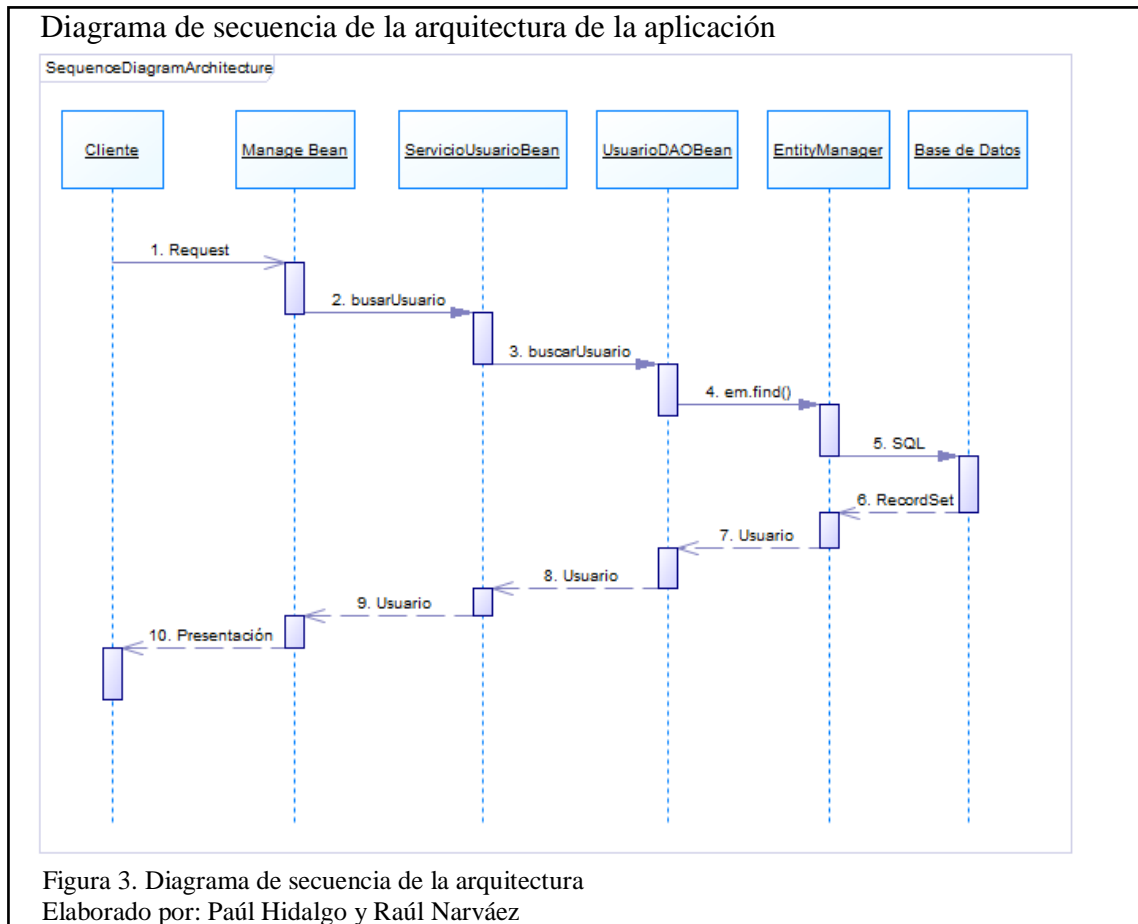
Todo el desarrollo está basado en componentes de negocio implementados en Enterprise JavaBeans (EJB), se permite una independencia a nivel de la base de datos, gracias a la utilización de un API de mapeo objetos-relacional (JPA), para la capa de presentación el modelo de componentes basado en JSF 2.0.

Modelo – Vista - Controlador

El modelo representa los datos de aplicación y las reglas que rigen el acceso y la modificación de dichos datos.

La vista presenta el contenido del modelo, esta despliega los datos del modelo y especifica cómo deben ser presentados los datos, cuando el modelo cambia, es responsabilidad de la vista mantener consistencia en la presentación La vista envía peticiones de usuario al controlador.

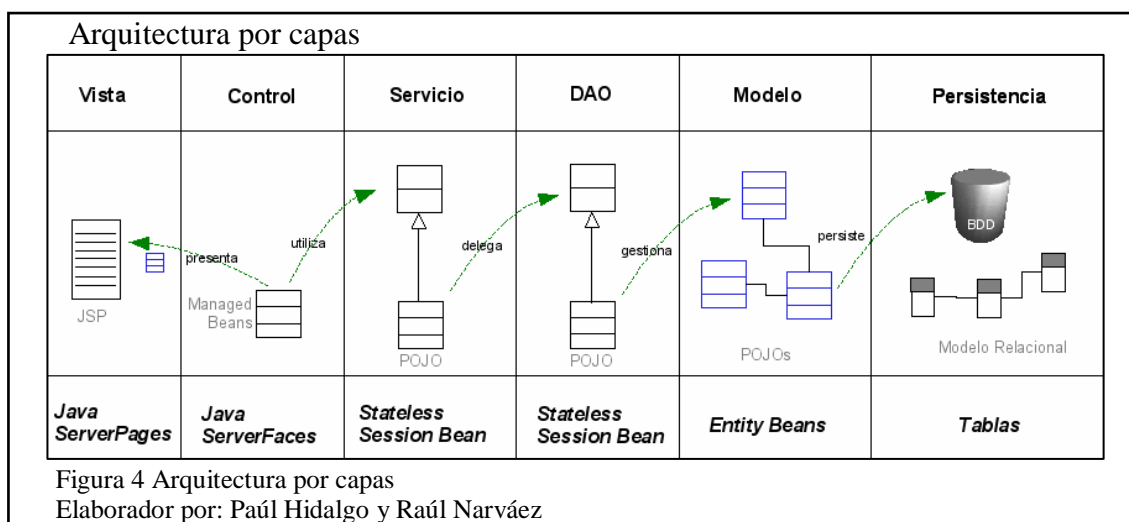
En el siguiente diagrama de secuencia muestra la cadena de invocación de un proceso de negocio, desde la capa de presentación hasta la persistencia:



Con respecto al cliente puede ser cualquier navegador o browser que realice las respectivas peticiones para realizar alguna acción, el controlador viene a ser el Managed Bean, mientras que el servicio de usuario en esta ocasión son las Session Bean, que finalmente derivan en las Entity Manager y la base de datos que son los que van a proveer la información solicitada desde el cliente.

1.2.9 Arquitectura por capas

La arquitectura de la presente aplicación está dividida en varias capas lógicas que poseen su correspondiente implementación usando la especificación JEE 7 quedando de la siguiente manera:



Persistencia

En esta capa se describe el mecanismo por el cual se obtiene y se ingresa información de la base de datos. En la especificación JPA la persistencia de datos es realizada mediante un mapeo Objetos- Relacional que nos permite almacenar un modelo de objetos en un modelo relacional utilizando tablas de base de datos, en forma transparente y sin tener que utilizar sentencias SQL.

Modelo

Representa todos los conceptos del dominio de negocio que necesitamos modelar dentro de la aplicación. La capa Modelo es fundamental en toda la arquitectura, puesto que son estos objetos que representan el estado del sistema los que serán llevados de capa en capa, hasta poder presentarlos al usuario.

La capa de modelo debe desarrollarse con Entity Beans de la especificación Enterprise JavaBeans 3.1, incluida en Java EE 7.

Data Access Objects (DAO)

Es una estrategia para la persistencia de datos. Los DAOs son los encargados de almacenar y obtener objetos de cualquier repositorio persistente de datos, como por ejemplo una base de datos relacional.

Estos DAOs son implementados utilizando el API de Persistencia de EJB 3.1, dentro de un Session Bean. Se ha implementado un Stateless Session Bean que centralice el acceso a datos aplicando el uso del API de Generics

Negocio

Define el comportamiento del sistema, y engloba todas las posibles funcionalidades del mismo. Es una buena práctica definir la capa de servicio de negocio como una interface, para desacoplar la estrategia de implementación de la definición de negocio.

La interface de Negocio no depende de la interface DAO. Esa dependencia se da en la implementación de la interface de Servicio, como parte de la estrategia de implementación.

Controlador

Gestiona las acciones visuales del XHTML con la ventaja de poder llamar a cualquier servicio ya que es un Session Bean basado en anotaciones.

Vista

Esta capa está formada por componentes JSF 2.0 dentro de páginas XHTML, es una cuestión de implementación que no afecta en la organización arquitectónica de la aplicación, ya que se siguen respetando los patrones y capas definidos anteriormente. Se ha utilizado “Contexts and Dependency Injection” (CDI) como mecanismo de inyección de dependencias entre JSF y EJB.

Finalmente, para mejorar la experiencia de los usuarios se sugiere utilizar PrimeFaces 5, que es compatible con JSF 2.0.

1.2.10 Versionamiento

Se debe contar con un mecanismo de control de versionamiento, y utilizarlo para el desarrollo de todas las aplicaciones.

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante. A pesar de que los ejemplos de este libro muestran código fuente como archivos bajo control de versiones, en realidad cualquier tipo de archivo que encuentres en un ordenador puede ponerse bajo control de versiones. (Git, 2017).

CAPÍTULO 2

ANÁLISIS Y DISEÑO

2.1 Análisis

Las empresas tienen la necesidad de llevar el control de sus empleados y para lograr este objetivo se han construido diferentes mecanismos ya sean estos netamente electrónicos, o combinaciones de aplicaciones de software con dispositivos electrónicos, además cada país tiene una serie de normas que se deben contemplar, lo que los obliga a llevar el control de los recursos humanos.

El tener definidos esquemas y estándares de programación es de gran utilidad para proceso de desarrollo del producto pues representa una ventaja al momento de la reutilización y la compartición del código. Los esquemas descritos en el presente capítulo están basados en el proceso unificado y UML adaptados a la web, puesto que es la definición que propone UWE.

2.2 Requerimientos del sistema

La especificación de requerimientos se la llevo a cabo de una investigación previa del mercado y una consulta al ministerio de relaciones laborales en relación con las funcionalidades que debería tener el procesamiento de la información relacionada con el control de la asistencia de recurso humano de la empresa.

El conjunto de funcionalidades que se van a implementar se encuentran registradas en la Tabla 2 (Product backlog).

Tabla 2. Requerimientos del sistema

MÓDULO	ITERACIÓN	ID HU	REQUERIMIENTOS	ID TAREA	TAREA	PRIORIDAD	QUIÉN	ESTIMADO
Seguridades	1	1	Análisis y diseño de proceso.	1.1	Diagrama de procesos	ALTO	PHIDALGO	3
				1.2	Diseño de modelo E-R	ALTO	PHIDALGO	3
				1.3	Creación de arquitectura	ALTO	PHIDALGO	2
		2	Gestión de perfiles	2.1	Crear Pantalla F-E de Gestión de perfiles.	ALTO	RNARVAEZ	4
				2.2	Codificación de funcionalidad de gestión de perfiles.	ALTO	RNARVAEZ	6
				2.3	Pruebas unitarias	MEDIA	RNARVAEZ	2
				2.4	Correcciones	BAJA	RNARVAEZ	3
		3	Gestión de módulos	3.1	Crear Pantalla F-E de Gestión de módulo.	ALTO	PHIDALGO	4
				3.2	Codificación de funcionalidad de gestión de módulos.	ALTO	PHIDALGO	6
				3.3	Pruebas unitarias	MEDIA	PHIDALGO	2
				3.4	Correcciones	BAJA	PHIDALGO	3

		4	Gestión de tipo de opciones	4.1	Crear Pantalla F-E de Gestión de tipo de opciones.	ALTO	RNARVAEZ	4
				4.2	Codificación de funcionalidad de gestión de tipo de opciones.	ALTO	RNARVAEZ	6
				4.3	Pruebas unitarias	MEDIA	RNARVAEZ	2
				4.4	Correcciones	BAJA	RNARVAEZ	3
	2	5	Análisis y diseño de proceso.	5.1	Diagrama de procesos	ALTO	PHIDALGO	3
				5.2	Diseño de modelo E-R	ALTO	PHIDALGO	3
				5.3	Creación de arquitectura	ALTO	PHIDALGO	2
		6	Gestión de permisos	6.1	Crear Pantalla F-E de Gestión de permisos.	ALTO	RNARVAEZ	4
				6.2	Codificación de funcionalidad de gestión de permisos.	ALTO	RNARVAEZ	6
				6.3	Pruebas unitarias	MEDIA	RNARVAEZ	2
				6.4	Correcciones	BAJA	RNARVAEZ	3
		7	Gestión de opciones	7.1	Crear Pantalla F-E de Gestión de opciones.	ALTO	PHIDALGO	4

Nómina	3			7.2	Codificación de funcionalidad de gestión de opciones.	ALTO	PHIDALGO	6
				7.3	Validación asignación de perfiles a opciones	MEDIA	PHIDALGO	4
				7.4	Pruebas unitarias	MEDIA	PHIDALGO	2
				7.5	Correcciones	BAJA	PHIDALGO	3
		8	Gestión de horarios	8.1	Crear Pantalla F-E de Gestión de horarios.	ALTO	RNARVAEZ	4
				8.2	Codificación de funcionalidad para horarios		RNARVAEZ	6
				8.2	Codificación de funcionalidad para detalle de horario	ALTO	RNARVAEZ	6
				8.3	Pruebas unitarias	MEDIA	RNARVAEZ	2
				8.4	Correcciones	BAJA	RNARVAEZ	4
		9	Planificación de festividades	9.1	Crear Pantalla F-E de planificación de festividades.	ALTO	PHIDALGO	4
				9.2	Codificación de funcionalidad de	ALTO	PHIDALGO	6

					planificación de festividades.			
				9.3	Pruebas unitarias	MEDIA	PHIDALGO	2
				9.4	Correcciones	BAJA	PHIDALGO	4
		10	Administración de empleados	10.1	Crear Pantalla F-E de Administración de empleados.	ALTO	RNARVAEZ	4
				10.2	Crear Pantalla F-E de Asignar perfiles.	ALTO	RNARVAEZ	4
				10.3	Crear Pantalla F-E de Asignar horario.	ALTO	RNARVAEZ	4
				10.4	Crear Pantalla F-E de Configuración de solicitudes.	ALTO	RNARVAEZ	4
				10.5	Codificación de funcionalidad de empleados.	ALTO	RNARVAEZ	6
				10.6	Codificación de funcionalidad de asignación de perfil.	ALTO	RNARVAEZ	4
				10.7	Codificación de funcionalidad de asignación de horario.	ALTO	RNARVAEZ	4

				10.8	Codificación de funcionalidad de configuración de solicitudes.	ALTO	RNARVAEZ	6
				10.9	Pruebas unitarias	ALTO	RNARVAEZ	2
				10.10	Correcciones	BAJA	RNARVAEZ	5
	4	11	Registro de asistencia manual	11.1	Crear Pantalla F-E de asistencia manual.	ALTO	PHIDALGO	4
				11.2	Codificación de funcionalidad de asistencia manual.	ALTO	PHIDALGO	6
				11.3	Pruebas unitarias	MEDIA	PHIDALGO	2
				11.4	Correcciones	BAJA	PHIDALGO	4
		12	Carga de información	12.1	Crear Pantalla F-E de carga de información	ALTO	PHIDALGO	4
				12.2	Codificación de funcionalidad de carga de información.	ALTO	PHIDALGO	6
				12.3	Pruebas unitarias	MEDIA	PHIDALGO	2
				12.4	Correcciones	BAJA	PHIDALGO	4
		13	Registro de actividades para horas extra	13.1	Crear Pantalla F-E de registro	ALTO	RNARVAEZ	4

					de actividades para hora extra.			
				13.2	Codificación de funcionalidad de registro de actividades para hora extra.	ALTO	RNARVAEZ	6
				13.3	Pruebas unitarias	MEDIA	RNARVAEZ	4
				13.4	Correcciones	BAJA	RNARVAEZ	4
		14	Generación de solicitudes	14.1	Crear Pantalla F-E de generación de solicitudes	ALTO	RNARVAEZ	4
				14.2	Codificación de funcionalidad de generación de solicitudes	ALTO	RNARVAEZ	6
				14.3	Pruebas unitarias	MEDIA	RNARVAEZ	2
				14.4	Correcciones	BAJA	RNARVAEZ	4
		15	Validación de solicitudes	15.1	Crear Pantalla F-E de validación de solicitudes.	ALTO	RNARVAEZ	4
				15.2	Codificación de funcionalidad de validación de solicitudes.	ALTO	RNARVAEZ	6
				15.3	Pruebas unitarias	MEDIA	RNARVAEZ	2

				15.4	Correcciones	BAJA	RNARVAEZ	4
	5	16	Reporte personal	16.1	Crear Pantalla F-E de reporte personal.	ALTO	PHIDALGO	4
				16.2	Codificación de funcionalidad de reporte personal.	ALTO	PHIDALGO	4
				16.3	Pruebas unitarias	MEDIA	PHIDALGO	2
				16.4	Correcciones	BAJA	PHIDALGO	4
		17	Reporte administrativo	17.1	Crear Pantalla F-E de reporte administrativo	ALTO	RNARVAEZ	4
				17.2	Codificación de funcionalidad de reporte administrativo.	ALTO	RNARVAEZ	4
				17.3	Pruebas unitarias	MEDIA	RNARVAEZ	2
				17.4	Correcciones	BAJA	RNARVAEZ	4
		18	Reporte gerenciales	18.1	Crear Pantalla F-E de reporte gerencial	ALTO	PHIDALGO	4
				18.2	Codificación de funcionalidad de reporte gerencial.	ALTO	PHIDALGO	4
				18.3	Pruebas unitarias	MEDIA	PHIDALGO	4

Nota: Tabla completa con los requerimientos del sistema de los 5 sprints
Elaborador por: Paúl Hidalgo y Raúl Narváez

2.2.1 Pruebas por iteración

Iteración 1: Se encuentra descrito el módulo de gestión de seguridades con los procesos de administración de perfiles, administración de módulos, y gestión de tipo de opciones teniendo un resultado aceptado por el usuario

Tabla 3. Resultado iteración 1

MÓDULO	ITERACIÓN	PRE REQUISITO	ACTOR	PROCESO	SUB-PROCESO	TAREA	RESULTADO ESPERADO	RESULTADO ALCANZADO	OBSERVACIONES
GESTIÓN DE SEGURIDADES	1	Acceso a sistema mediante usuario y contraseña.	Recursos humanos	Administración de perfiles	Gestión de Perfiles	Mostrar listado de detalles de perfiles.	Figura anexo 1	OK	
						Eliminar perfil seleccionado.			
						Crear un perfil.	Figura anexo 2	OK	
						Modificar datos del perfil.			
				Administración de módulos	Gestión de módulos	Mostrar listado de módulos.	Figura anexo 3	OK	
						Eliminar módulo			
						Crear módulo	Figura anexo 4	OK	

						Modificar módulo			
				Administración de tipo de opciones	Gestión de tipo de opciones	Mostrar listado de tipo de opciones		NO COMPLETADO	NO SE ENCUENTRA REFRESCANDO DEBIDAMENTE LA GRILLA
						Eliminar de tipo de opciones			
						Registro de un nuevo tipo de opción	Figura anexo 5	OK	
						Modificar tipo de opciones			

Nota: Resultado de la entrega del sprint 1

Elaborador por: Paúl Hidalgo y Raúl Narváez

Iteración 2: Se encuentra descrito el módulo de gestión de seguridades con los procesos de administración de tipos de opciones con el respectivo arreglo acerca de la observación descubierta en el entregable del sprint 1, administración de permisos, y gestión de opciones teniendo un resultado aceptado por el usuario.

Tabla 4. Resultado iteración 2

MÓDULO	ITERACIÓN	PRE REQUISITO	ACTOR	PROCESO	SUB-PROCESO	TAREA	RESULTADO ESPERADO	RESULTADO ALCANZADO	OBSERVACIONES
GESTIÓN DE SEGURIDADES	2	Acceso a sistema mediante usuario y contraseña.	Recursos humanos	Administración de tipo de opciones	Gestión de tipo de opciones	Mostrar listado de tipo de opciones	Figura anexo 6	OK	CORRECCIÓN REALIZADA
				Administración de permisos	Gestión de permisos	Mostrar listado de permisos.	Figura anexo 7	OK	
						Eliminar permiso seleccionado.	Figura anexo 8	OK	
						Crear un permiso.			
						Modificar datos del permiso.			
				Administración de opciones	Gestión de opciones	Mostrar listado de opciones.	Figura anexo 9	OK	
						Eliminar opciones			
						Crear opciones	Figura anexo 10	OK	
						Modificar opciones			

Nota: Resultado de la entrega del sprint 2

Elaborado por: Paúl Hidalgo y Raúl Narváez

Iteración 3: Se encuentra descrito el módulo de gestión de recurso de nómina con los procesos de administración de horarios, teniendo un resultado aceptado por el usuario, planificación de festividades, administración de empleados teniendo un resultado aceptado por el usuario.

Tabla 5. Resultado iteración 3

MÓDULO	ITERACIÓN	PRE REQUISITO	ACTOR	PROCESO	SUB-PROCESO	TAREA	RESULTADO ESPERADO	RESULTADO ALCANZADO	OBSERVACIONES
Gestión de recursos de nomina	3	Acceso a sistema mediante usuario y contraseña.	Recursos humanos	Administración de horarios	Gestión de Detalle de horario	Mostrar listado de detalles de horario.	Figura anexo 11	OK	
						Eliminar Detalle horario seleccionado.			
						Crear un detalle de horario.	Figura anexo 12	OK	
						Modificar datos del detalle horario			
					Gestión de Horarios	Mostrar listado de horarios.	Figura anexo 13	OK	
						Eliminar horario seleccionado.			
						Crear un horario	Figura anexo 14	OK	
						Modificar horario			
				Planificación de festividades	Gestión de días festivos	Mostrar listado de días festivos.	Figura anexo 15	OK	

						Eliminar día festivo			
						Crear día festivo	Figura anexo 16	OK	
						Modificar día festivo			
			Administración de empleados	Gestión de empleados		Mostrar listado de empleados	Figura anexo 17	OK	
						Eliminar de empleado seleccionado.			
						Registro de un nuevo empleado.	Figura anexo 18	NO COMPLETADO	Registro de imagen para uso en lector facial. Pendiente por definir proceso de asistencia.
						Modificar de datos del empleado.		OK	
				Asignación de perfiles		Mostrar listado de perfiles asignados a empleado seleccionado.	Figura anexo 19	OK	
						Eliminar perfil de empleado seleccionado.			
						Asignar un perfil al empleado seleccionado.	Figura anexo 20	OK	

					Modificar perfiles asignados a empleado seleccionado.		OPTIMIZACIÓN PENDIENTE	Problema con componente pickList para la modificación de perfiles asignados. Pendiente para desarrollo en siguiente sprint
					Mostrar listado de horarios asignados a empleado seleccionado.	Figura anexo 21	OK	
					Eliminar horario de empleado seleccionado.			
				Asignación de horario	Asignar un horario al empleado seleccionado.	Figura anexo 22	OK	
					Modificar horarios asignados a empleado seleccionado.			
					Asignar vigencia de horarios extras	Figura anexo 23	OK	
				Configuración de solicitudes	Mostrar listado de	Figura anexo 24	OK	

						configuración de solicitudes del empleado seleccionado.			
						Eliminar configuración de solicitudes.			
						Crear una nueva configuración de solicitudes.			Generar una configuración exclusiva para cada tipo de solicitud. Pendiente para desarrollo en siguiente sprint
						Modificación de configuración de solicitudes	Figura anexo 25	OK	

Nota: Resultado de la entrega del sprint 3

Elaborado por: Paúl Hidalgo y Raúl Narváez

Iteración 4: Se encuentra descrito el módulo de gestión de recurso de nómina con los procesos de registro de asistencia, actividades personales y validar solicitudes teniendo un resultado aceptado por el usuario.

Tabla 6. Resultado iteración 4

MÓDULO	ITERACIÓN	PRE REQUISITO	ACTOR	PROCESO	SUB-PROCESO	TAREA	RESULTADO ESPERADO	RESULTADO ALCANZADO	OBSERVACIONES
Gestión de recursos de nomina	4	Acceso a sistema mediante usuario y contraseña.	Empleado	Registro de asistencia	Registro de asistencia manual	Mostrar listado de listado de registros de asistencia.	Figura anexo 26	OK	
						Eliminar registro de asistencia seleccionado.			
						Crear un registro de asistencia de forma manual.	Figura anexo 27	OK	
						Modificar registro de asistencia seleccionado.			
			Recursos humanos		Carga de información	Mostrar listado de asistencia de usuarios seleccionados en un rango de fechas.	Figura anexo 28	OK	
						Descarga de formato para subir datos.			
						Importar datos de archivo Excel.			

			Empleado	Actividades personales	Registro de actividades para horas extra	Mostrar listado de listado de actividades.	Figura anexo 29	OK	
						Eliminar actividad seleccionada.			
						Crear un una actividad en hora extra calculada.	Figura anexo 30	OK	
						Modificar actividad seleccionada.			
					Generación de solicitudes	Mostrar listado de solicitudes registradas.	Figura anexo 31	OK	
						Crear un una solicitud.	Figura anexo 32	OPTIMIZACIÓN PENDIENTE	Considerar un tiempo de gracia para considerar atraso a un registro.
						Modificar estado de solicitud.		OK	
			Empleado	Validar solicitudes	Aprobar o negar solicitudes	Mostrar listado solicitudes para validar.	Figura anexo 33	OK	
						Aprobar o negar solicitudes.		OPTIMIZACIÓN PENDIENTE	Para las solicitudes de horas extra se aprueba las actividades registradas y después enviar al jefe

									superior si es necesario.
--	--	--	--	--	--	--	--	--	---------------------------

Nota: Resultado de la entrega del sprint 4

Elaborado por: Paúl Hidalgo y Raúl Narváez

Iteración 5: Se encuentra descrito el módulo de gestión de recurso de nómina con los procesos de reportes y optimización de actividades pendientes de la iteración 3 y 4 teniendo un resultado aceptado por el usuario.

Tabla 7. Resultado iteración 5

MÓDULO	ITERACIÓN	PRE REQUISITO	ACTOR	PROCESO	SUB-PROCESO	TAREA	RESULTADO ESPERADO	RESULTADO ALCANZADO	OBSERVACIONES
Gestión de recursos de nomina	5	Acceso a sistema mediante usuario y contraseña.	Empleado	Reportes	Reporte personal	Mostrar listado de solicitudes registradas.	Figura anexo 34	OK	
						Generación de reporte en formato Excel.			
			Recursos humanos		Reporte administrativo	Mostrar listado de solicitudes de usuarios seleccionadas.	Figura anexo 35	OK	
						Generación de reporte en formato Excel.			

			Gerencia		Reporte gerenciales	Muestra Gráficos estadísticos según información registrada	Figura anexo 36	OK	
			Recursos humanos	Administración de empleados	Asignación de perfiles	Modificar perfiles asignados a empleado seleccionado.		OK	OPTIMIZACIÓN RELIZADA
					Configuración de solicitudes	Crear una nueva configuración de solicitudes.	Figura anexo 37	OK	OPTIMIZACIÓN RELIZADA
			Empleado	Actividades personales	Generación de solicitudes	Crear un una solicitud.	Figura anexo 38	OK	OPTIMIZACIÓN RELIZADA
				Validar solicitudes	Aprobar o negar solicitudes	Aprobar o negar solicitudes.	Figura anexo 39	OK	OPTIMIZACIÓN RELIZADA

Nota: Resultado de la entrega del sprint 5
Realizado por: Paúl Hidalgo y Raúl Narváez

2.2.1 Sprint Backlog

Se observa como se ha desarrollado el software conforme ha ido transcurriendo el proceso de desarrollo del proyecto, esto en cuanto a los módulos que están especificados en el mismo. Estimación de tiempo se determina el tiempo que toma desarrollar un incremento.

Tabla 8. Sprint Backlog

ITERACIÓN	REQUERIMIENTOS	ESTADO	FECHA COMPLETADO	ESTIMADO
1	Análisis y diseño de proceso.	Terminado	06/01/2017	82
	Gestión de perfiles	Terminado	13/01/2017	
	Gestión de módulos	Terminado	20/01/2017	
	Gestión de tipo de opciones	Pendiente	27/01/2017	
2	Análisis y diseño de proceso.	Terminado	11/02/2017	82
	Gestión de permisos	Terminado	18/02/2017	

	Gestión de opciones	Terminado	25/02/2017	
3	Gestión de horarios	Terminado	11/03/2017	82
	Planificación de festividades	Terminado	18/03/2017	
	Administración de empleados	Pendiente	31/03/2017	
4	Registro de asistencia manual	Terminado	08/04/2017	82
	Carga de información	Terminado	22/04/2017	

	Registro de actividades para horas extra	Pendiente	22/04/2017	
	Generación de solicitudes	Terminado	29/04/2017	
	Validación de solicitudes	Pendiente	29/04/2017	
5	Reporte personal	Terminado	12/05/2017	82
	Reporte administrativo	Terminado	19/05/2017	
	Reporte gerenciales	Terminado	26/05/2017	

Nota: Tabla describiendo el sprint backlog
Elaborado por: Paúl Hidalgo y Raúl Narváez

2.3 Diagrama de procesos

Se realizó el diagrama global del proceso de negocio debido que es útil para identificar las funcionalidades que van a ser parte del software y además es agrega un valor significativo para tener claro el camino que hay que seguir durante todo el proceso de desarrollo del software.

En la Figura 5 se puede observar el proceso global de la gestión de la información relacionada con el control de asistencia de recurso humano de la empresa, el cuál ayudo a identificar los procesos que se procedió a automatizar, además el mencionado

diagrama fue útil para identificar los actores involucrados: Administrador del software, Recursos Humanos, Empleado, Jefe de Aprobación; cabe recalcar que conforme se vaya avanzando el proyecto los artefactos irán sufriendo cambios para un mayor entendimiento del mismo, esto se debe a la naturaleza de la metodología de desarrollo de software SCRUM, que es la que se está usando.

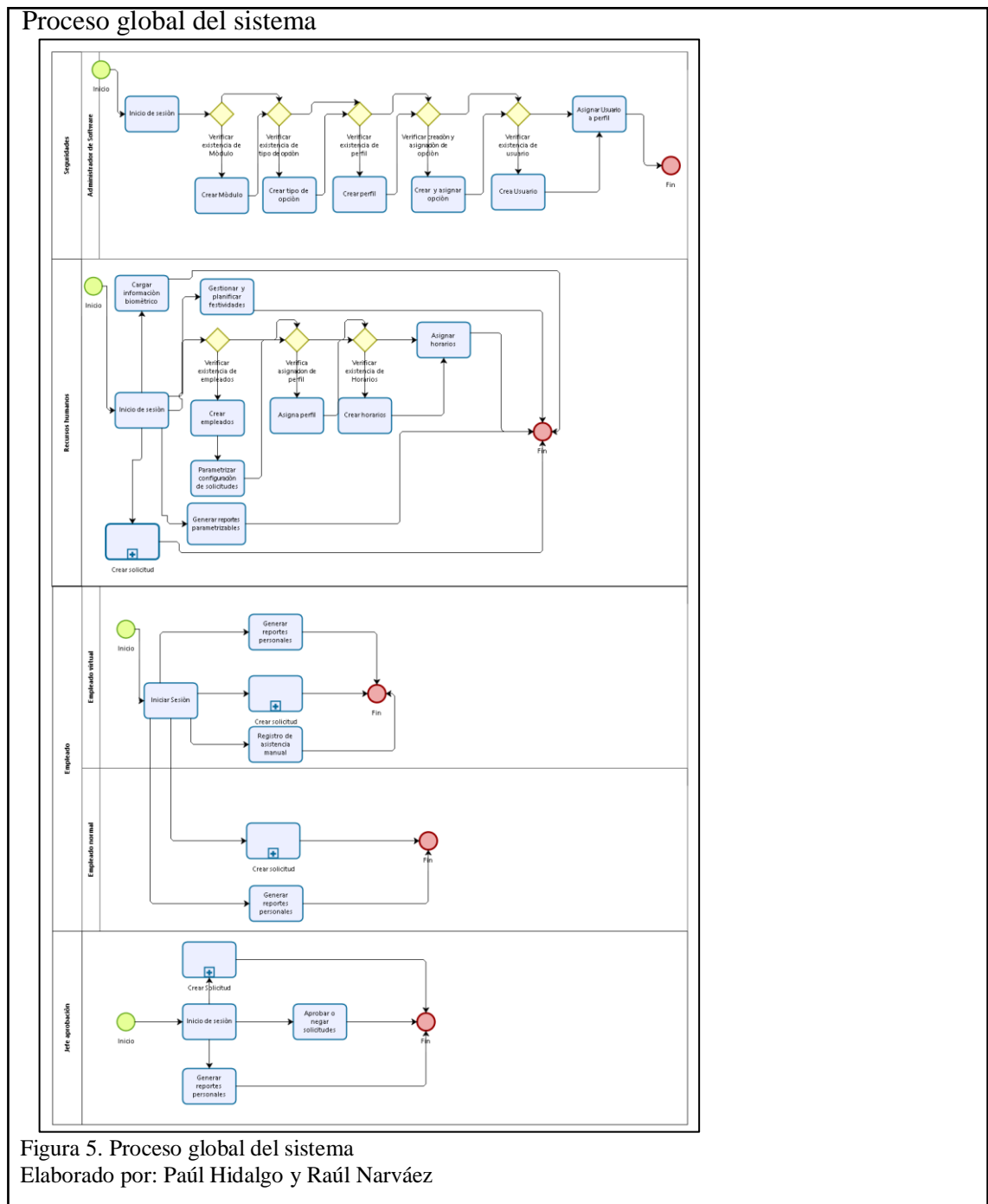


Figura 5. Proceso global del sistema
Elaborado por: Paúl Hidalgo y Raúl Narváez

2.4 Diagramas de casos de uso

Los diagramas de caso de uso son un método práctico para explorar requerimientos y permiten tener un conocimiento claro del funcionamiento del producto

Un caso de uso es una forma de expresar cómo alguien o algo externo a un sistema lo usa. Cuando decimos “alguien o algo” hacemos referencia a que los sistemas son usados no sólo por personas, sino también por otros sistemas de hardware y software. (Ceria, 2001, pág. 1).

En la Tabla 3 se compone de los campos: Rol y Tarea, siendo el primero los actores que van a formar parte para el sistema informático y el segundo una descripción de lo que realiza cada actor identificado.

Tabla 9. Actores del sistema y su descripción

ROL	TAREA
Administrador	Usuario encargado de la administración del sistema, tiene acceso a todos los módulos que son parte del sistema, y es el encargado de realizar las configuraciones necesarias para que el software mantenga una funcionalidad correcta
Recursos humanos	Usuario encargado de cargar información a partir de un biométrico, gestiona y planifica festividades, crea empleados y asigna horarios según el perfil con el que ingrese, parametriza la configuración de solicitudes
Empleado	Es el encargado de registrar la asistencia manual en el caso de ser necesario y crea las diferentes solicitudes a las que se la ha dado el acceso
Jefe de aprobación	Usuario encargado de aprobar o negar solicitudes, además de la generación de reportes.

Nota: Actores del sistema y la descripción correspondiente
Elaborado por: Paúl Hidalgo y Raúl Narváez

En la siguiente parte se describirán los casos de uso del sistema informático, cada uno conformado por el respectivo nombre, los actores que están involucrados, el curso

normal constituido por la secuencia de pasos que siga el actor por el sistema y la alternativa, que son las opciones a seguir dentro del sistema.

2.4.1 Configurar sistema

Proceso que el administrador del software realiza para gestionar los módulos del sistema, los tipos de opciones, perfiles, usuarios y la asignación de usuarios a los perfiles, además de las opciones a los perfiles.

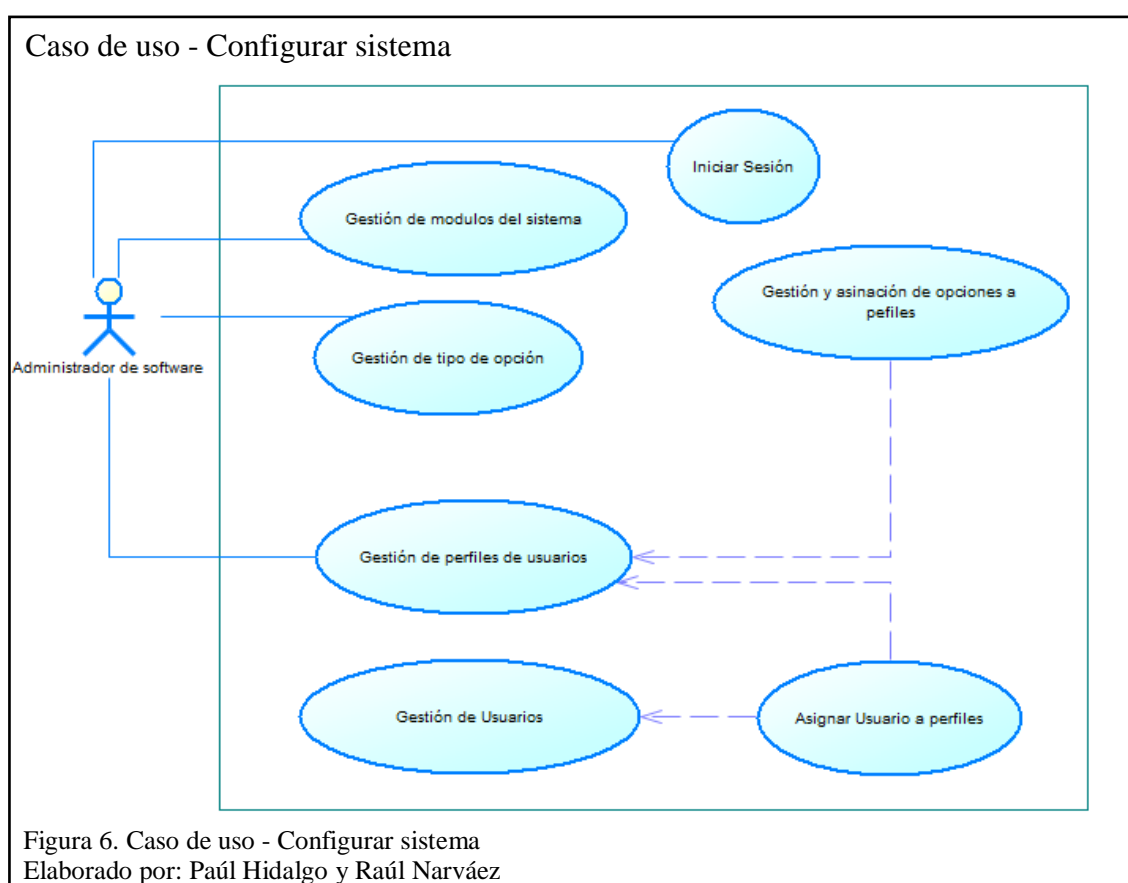


Tabla 10. Configurar sistema

Caso de uso: Configurar sistema	
Actores: Administrador	
Curso normal	Alternativa
1. Iniciar sesión	
2. Gestión de módulos del sistema	
3. Gestión de tipo de opción	

4. Gestión de perfiles de usuarios	4.1 Gestión y asignación de opciones a perfiles 4.2 Asignar usuario a perfiles
5. Gestión de usuarios	

Nota: Tabla del caso de uso de la configuración del sistema
Elaborado por: Paúl Hidalgo y Raúl Narváez

2.4.2 Registro y revisión de solicitudes

Proceso en el que el empleado registra las actividades, crea nuevas solicitudes y saca reportes personales, una vez generada las solicitudes tienen la potestad de validar, aprobar o negar las solicitudes el gerente y el administrador de recursos humanos, sólo el administrador de recursos humanos se encarga de las actividades personales.

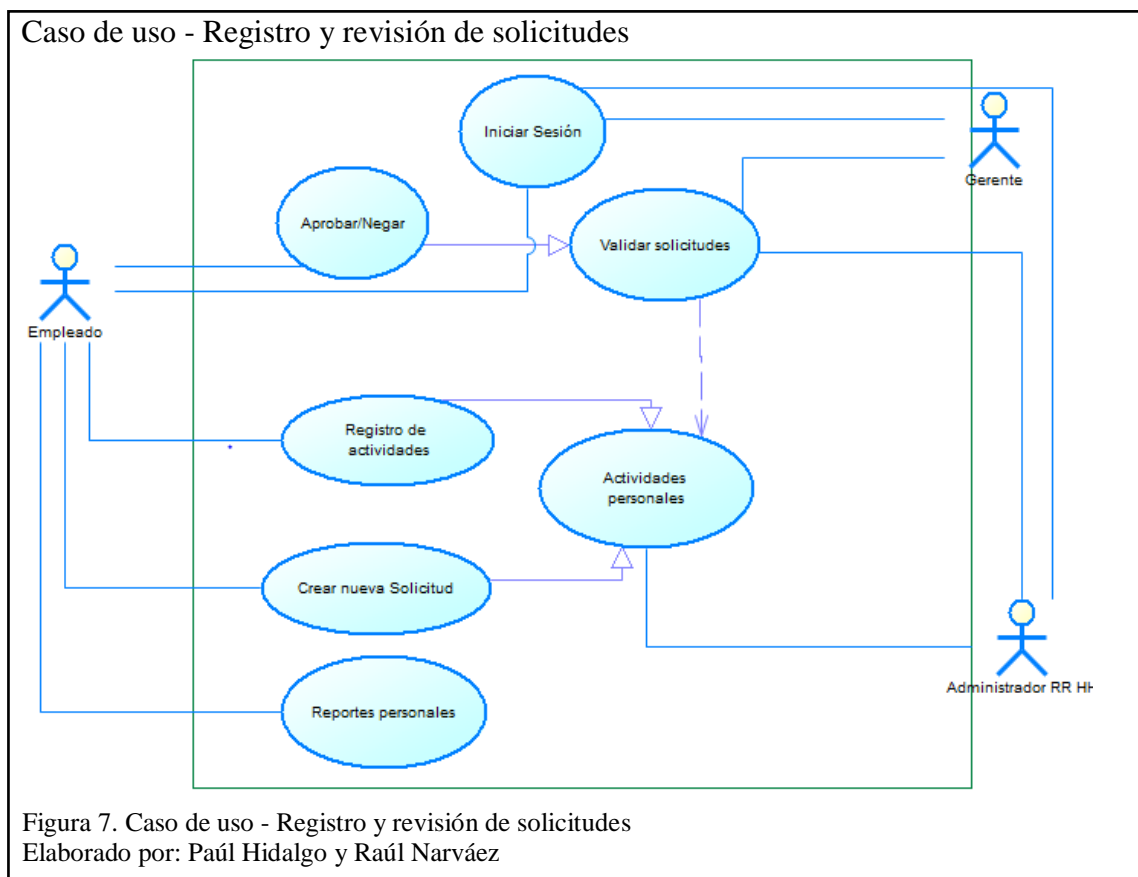


Tabla 11. Registro y revisión de solicitudes

Caso de uso: Registro y revisión de solicitudes	
Actores: Empleado, gerente y administrador de recursos humanos	
Curso normal	Alternativa
1. Iniciar sesión	
2. Registrar actividades	
3. Actividades Personales	3.1 Validar solicitudes
4. Crear nuevas solicitudes	
5. Aprobar/Negar solicitudes	
6. Validar solicitudes	
7. Reportes personales	

Nota: Tabla del registro y revisión del caso de uso de la revisión de solicitudes

Elaborado por: Paúl Hidalgo y Raúl Narváez

2.4.3 Tareas gerenciales

Proceso en el que el empleado genera las solicitudes mientras que el gerente se encarga de aprobar o negar y validar dichas solicitudes, además de generar reportes gerenciales, por otro lado, el administrador de recursos humanos solo valida las solicitudes y genera solicitudes.

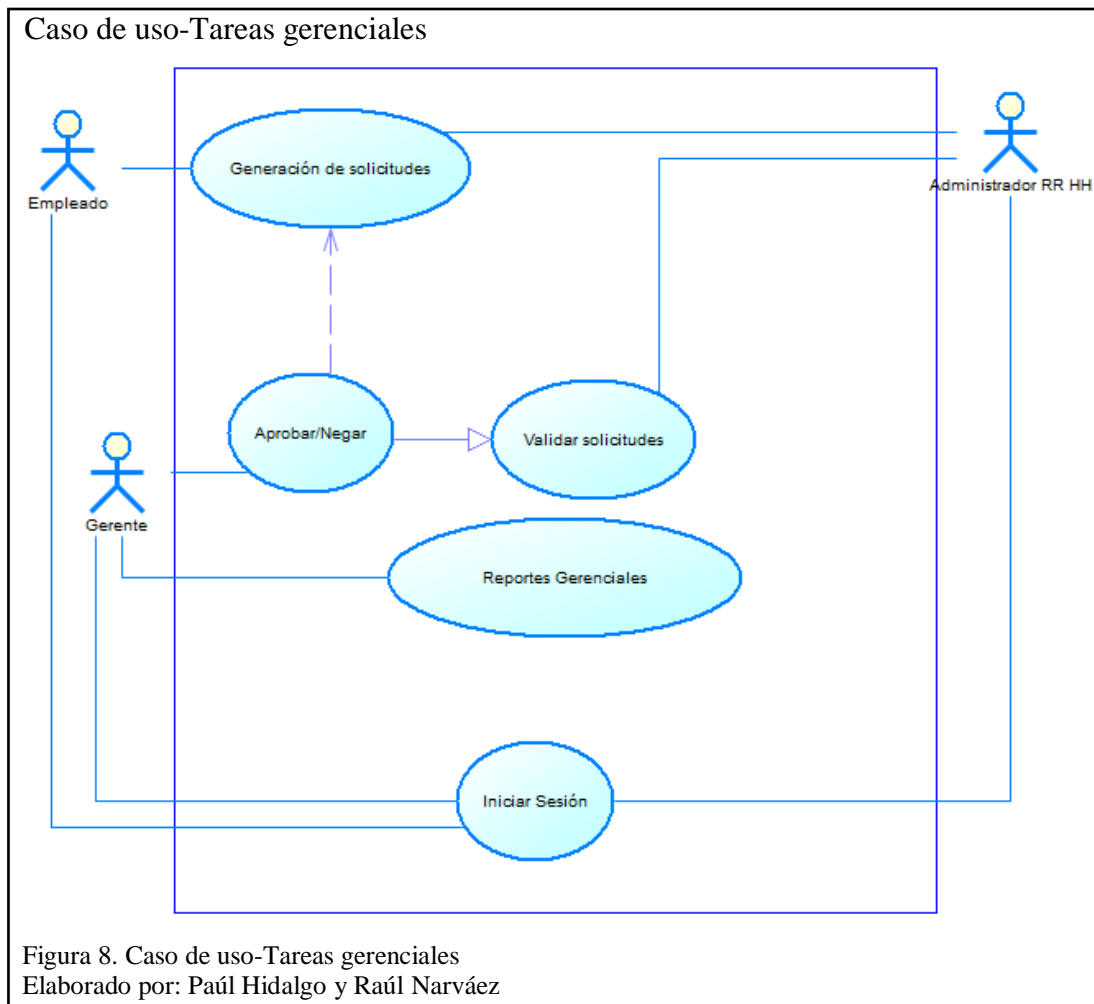


Tabla 12. Tareas gerenciales

Caso de uso: Tareas gerenciales	
Actores: Empleado, gerente y administrador de recursos humanos	
Curso normal	Alternativa
1. Iniciar sesión	
2. Generación de solicitudes	
3. Aprobar/Negar solicitudes	3.1 Generación de solicitudes
4. Validar solicitudes	
5. Reportes gerenciales	

Nota: Descripción del caso de uso de las tareas gerenciales

Elaborado por: Paúl Hidalgo y Raúl Narváez

2.4.4 Administrador de información

Proceso en el que el administrador de recursos humanos carga la asistencia de empleados, gestiona y planifica festividades, gestiona la jornada laboral y construye el horario laboral, además de la gestión de empleados que contempla la asignación de perfiles, horarios laborales y envío de solicitudes

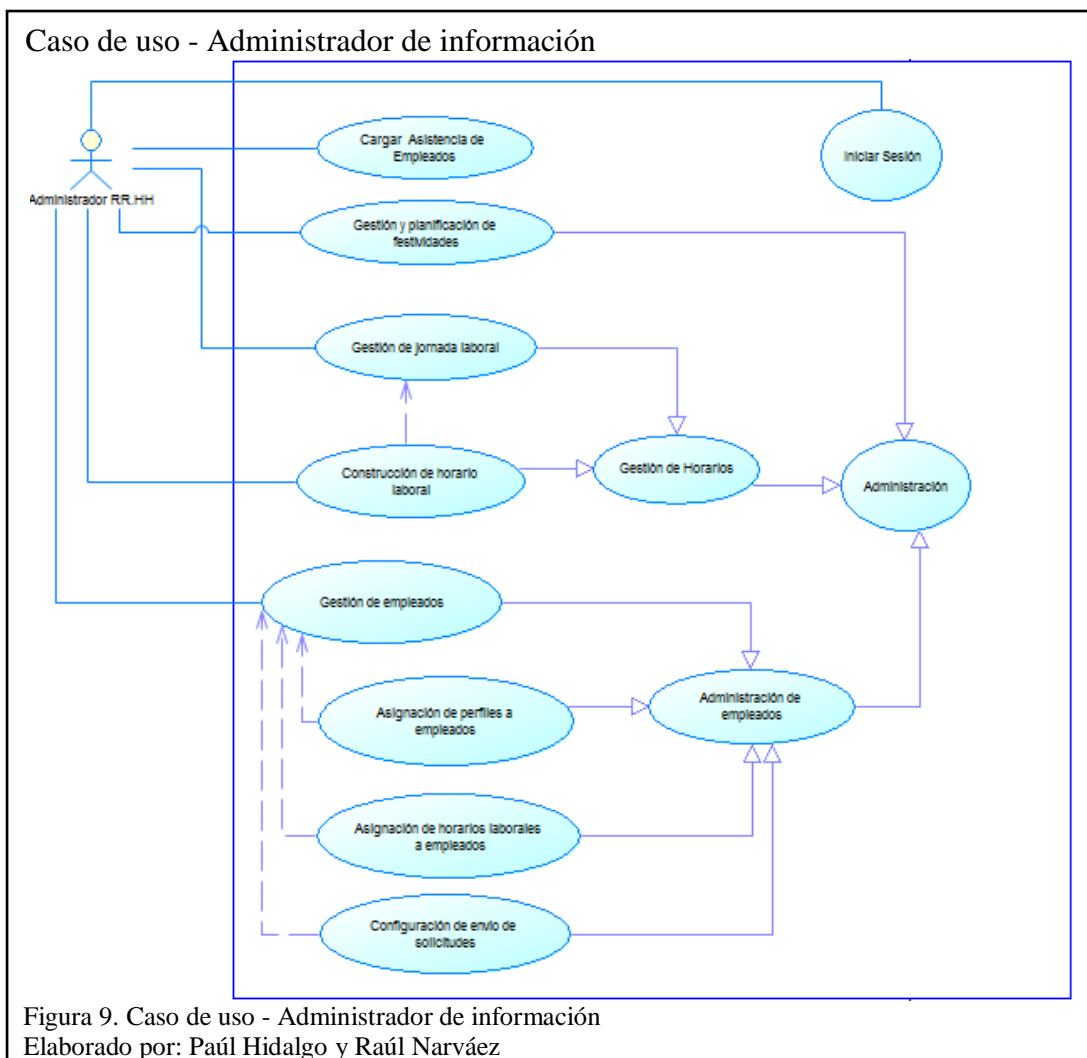


Tabla 13. Administrador de información

Caso de uso: Administrador de información	
Actores: Administrador de recursos humanos	
Curso normal	Alternativa
1. Iniciar sesión	
2. Cargar asistencia de empleados	

3. Gestión y planificación de festividades	
4. Gestión de jornada laboral	
5. Construcción de horario laboral	
6. Gestión de empleados	6.1 Asignación de perfiles a empleados 6.2 Asignación de horarios laborales a empleados 6.3 Configuración de envío de solicitudes

Nota: Descripción del caso de uso del administrador de recursos humanos
 Elaborado por: Paúl Hidalgo y Raúl Narváez

2.5 Diagramas de secuencia

Permiten tener una visión más clara del comportamiento de sistema en sus diferentes módulos, por la metodología de desarrollo usada cabe recalcar que sufrió cambios a medida que avanzaba el proyecto.

2.5.1 Administración de empleados

En la siguiente Figura se presenta el proceso de como el actor en este caso el administrador de recursos humanos interactúa con el sistema teniendo entre sus principales actividades el ingreso de nuevos empleados, asignación de perfiles y horarios, además de la edición de datos del empleado como de la configuración del mismo.

Diagrama de secuencia - Administración de empleados

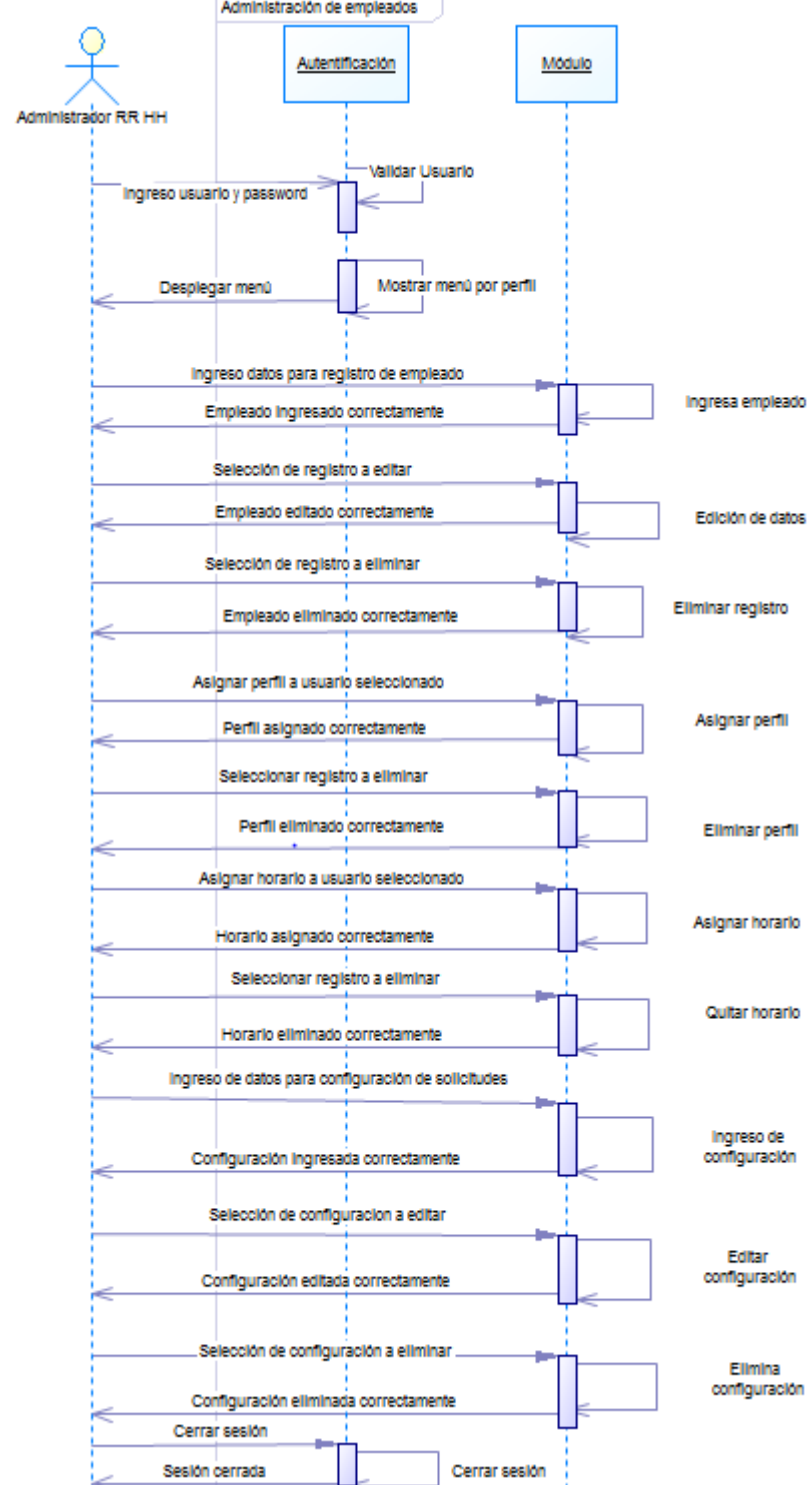
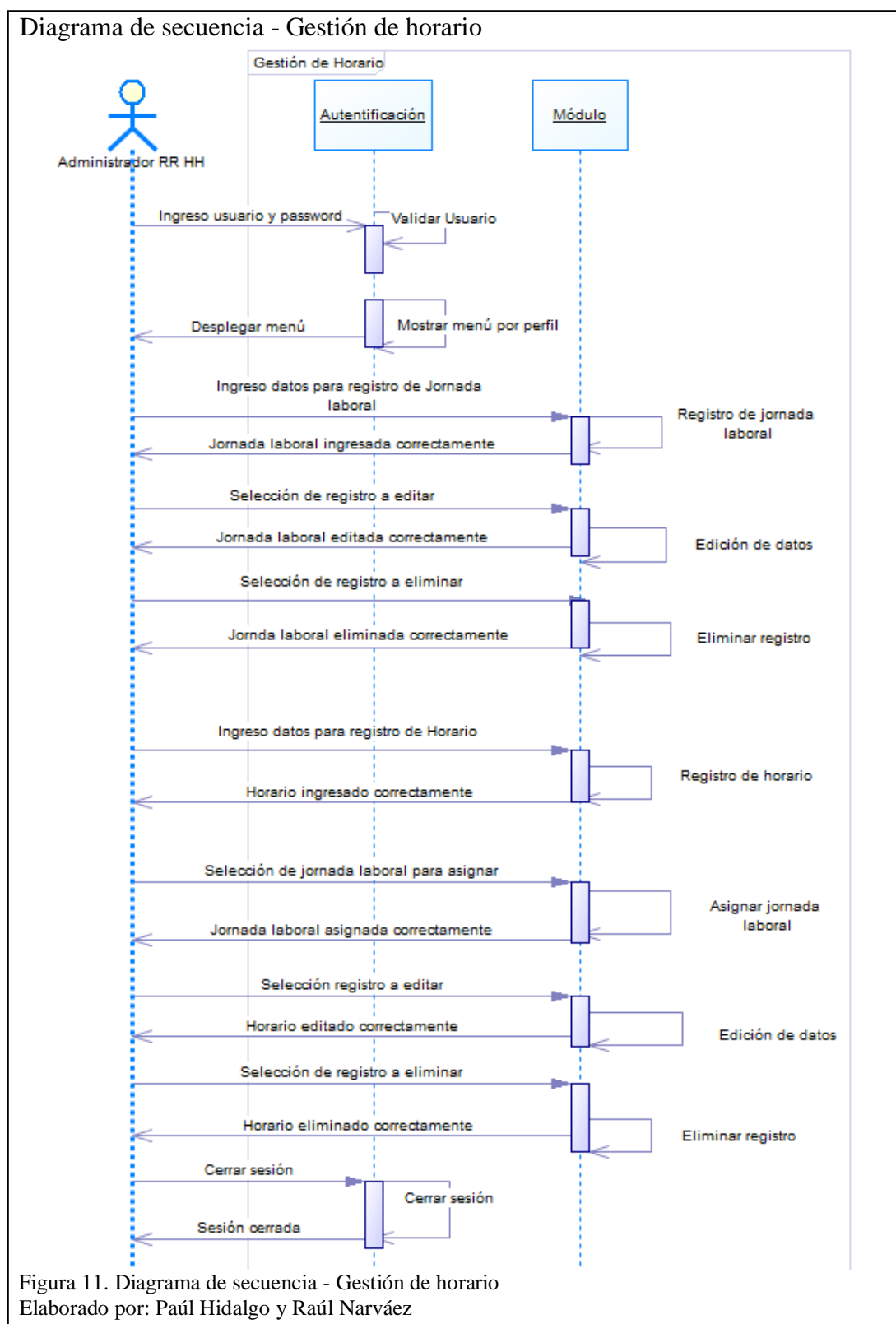


Figura 10. Diagrama de secuencia - Administración de empleados
Elaborado por: Paúl Hidalgo y Raúl Narváez

2.5.2 Gestión de horario

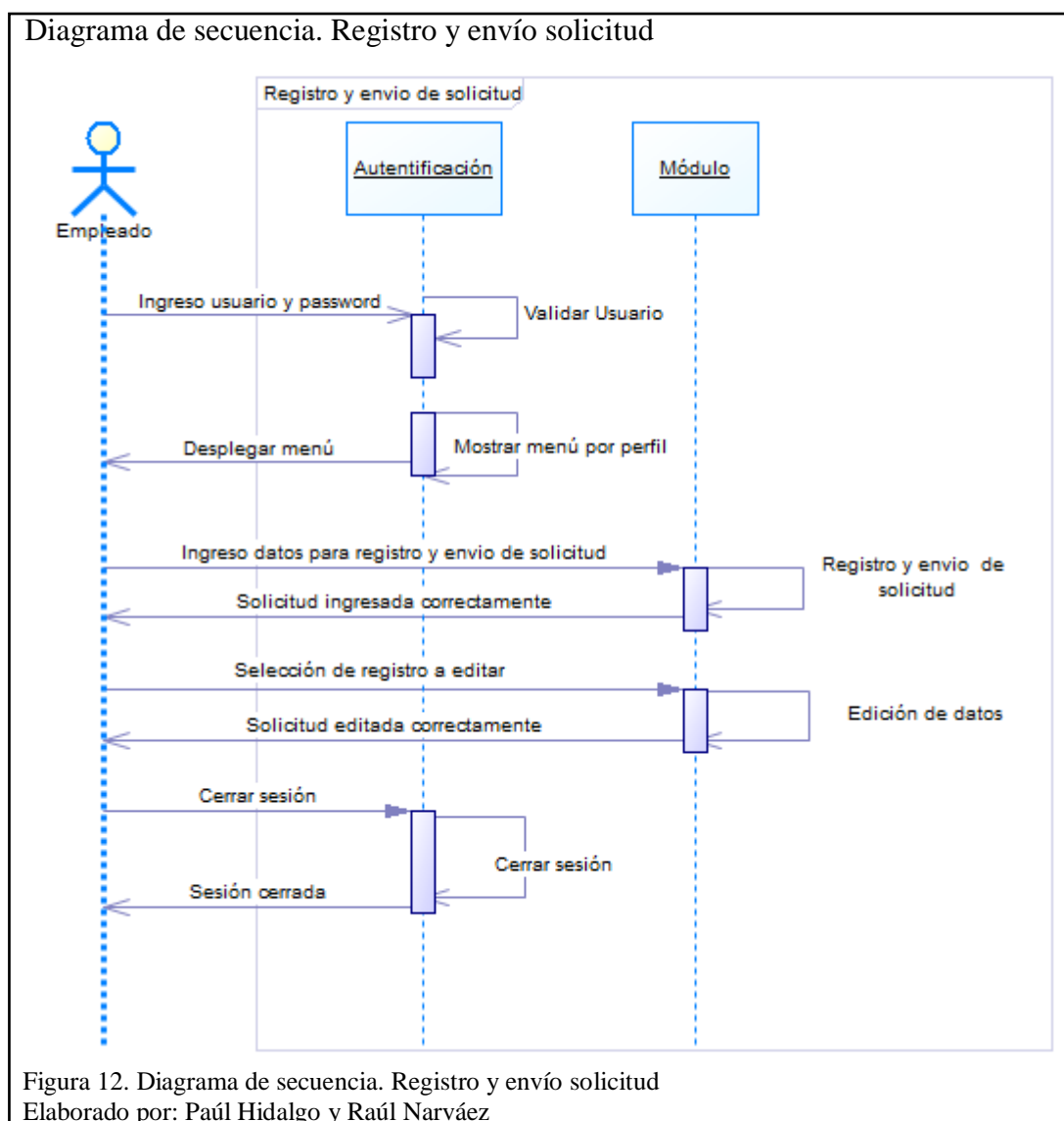
En la siguiente Figura se describe el proceso de como el actor en este caso el administrador de recursos humanos interactúa con el sistema con respecto a la gestión

del horario, teniendo entre sus actividades principales el registro de la jornada laboral y horario, además de la asignación de la jornada laboral.



2.5.3 Registro y envío de solicitud

En la siguiente figura se describe la interacción del actor en esta ocasión el empleado con la aplicación, destacando entre las principales actividades la validación de usuario, el registro y envío de solicitudes.



2.5.4 Planificación de festividades

En la siguiente Figura se describe el proceso que el actor, en este caso el administrador de recursos humanos, sigue al momento de planificar las festividades, teniendo como principales actividades el registro de la festividad, la edición o eliminación de cada uno de ellos.

Diagrama de secuencia. Planificación de festividades

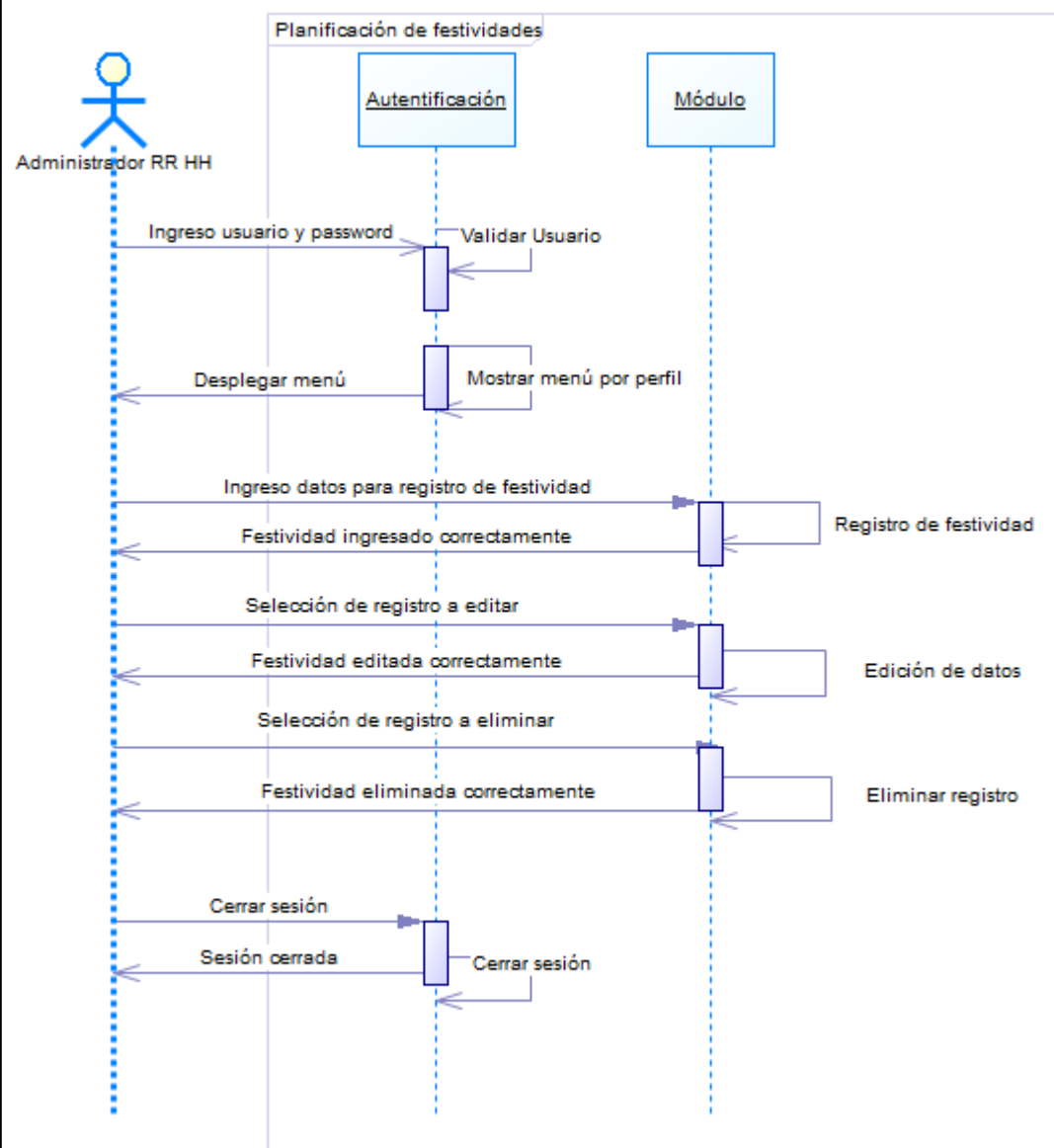
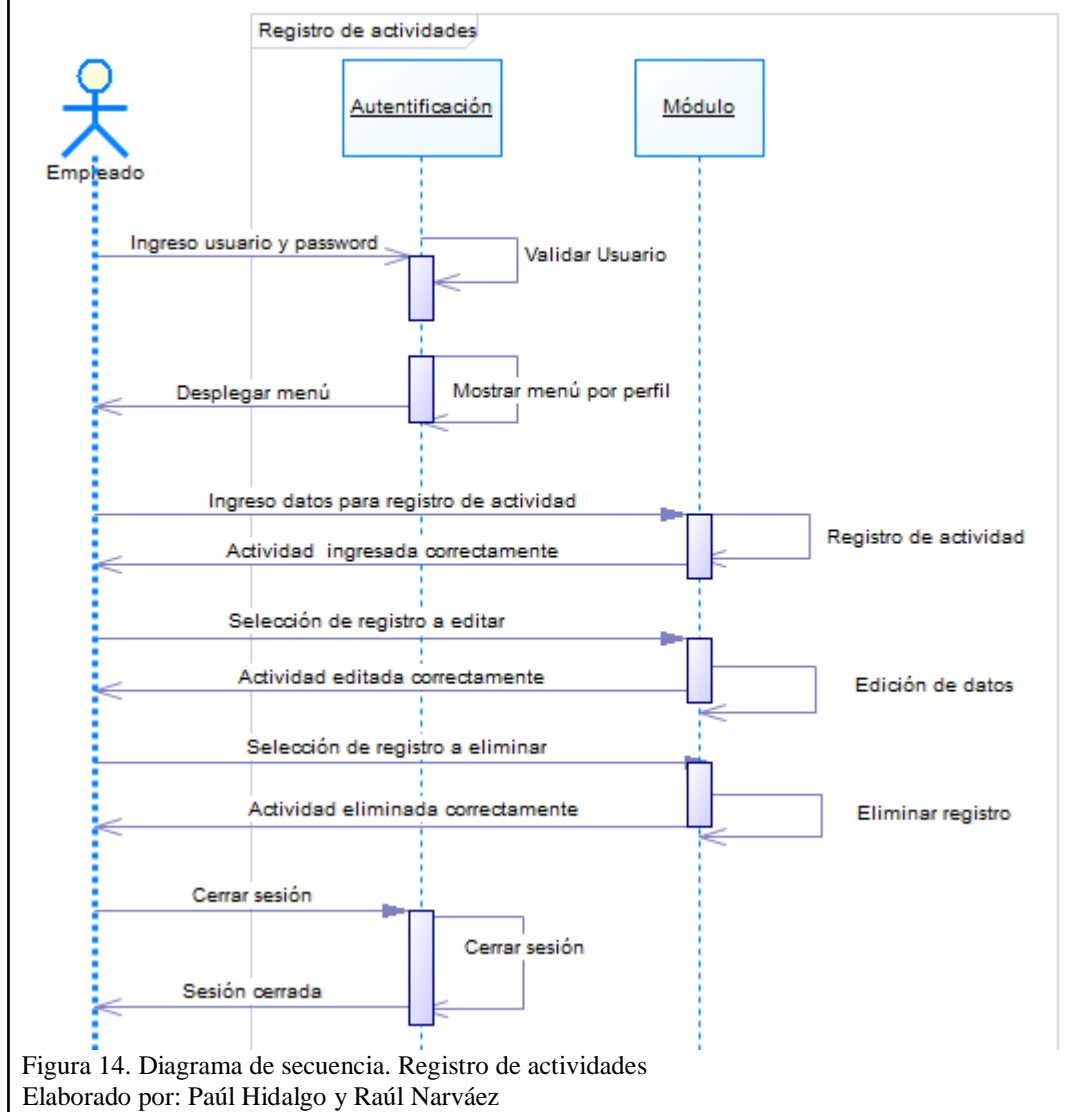


Figura 13. Diagrama de secuencia. Planificación de festividades
Elaborado por: Paúl Hidalgo y Raúl Narváez

2.5.5 Registro de actividades

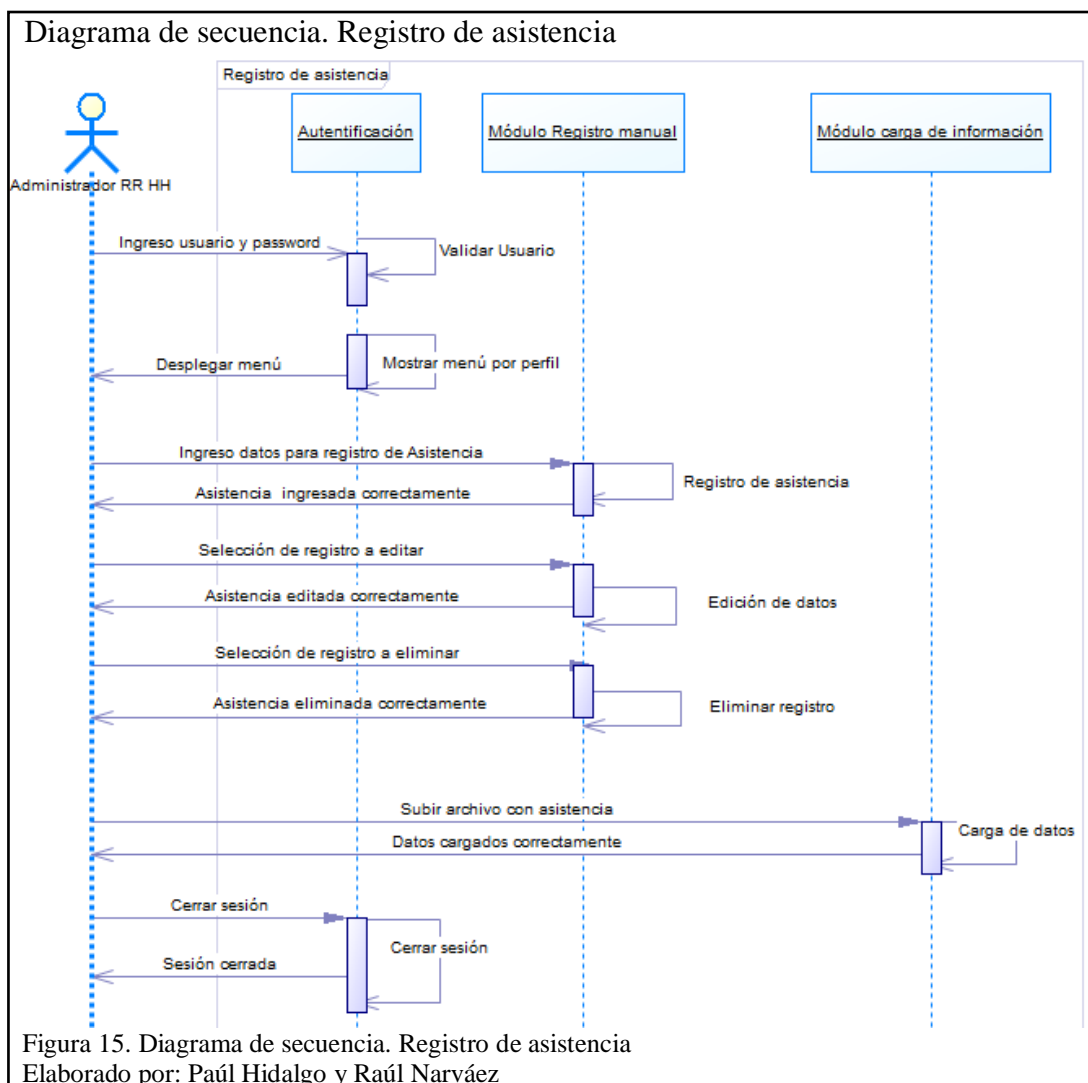
En la siguiente Figura se muestra el proceso que el actor en este caso el empleado realiza el proceso de registrar las actividades que realiza en su jornada laboral, teniendo entre sus principales funciones el registro de las actividades antes mencionadas, la edición y eliminación.

Diagrama de secuencia. Registro de actividades



2.5.6 Registro de asistencia

En la siguiente Figura se describe el proceso que el actor en esta ocasión el administrador de recursos humanos realiza al momento de registrar la asistencia del personal, teniendo entre sus primeras actividades el registro de la asistencia antes mencionada por cada empleado, además de la carga de datos que se lo realizaría de una manera automática a partir de reporte de Excel generado por el biométrico.



2.5.7 Validar solicitud

En la siguiente Figura se describe el proceso que el actor en este caso el empleado realiza la validación de solicitudes, ya que el mismo empleado puede ser jefe de otro dependiendo de la configuración que se le haya asignado en un principio, además entre las principales actividades que va a realizar son la aprobación o negación de las solicitudes y a su vez el envío a un superior en el caso de que exista.

Diagrama de secuencia. Validación de solicitudes

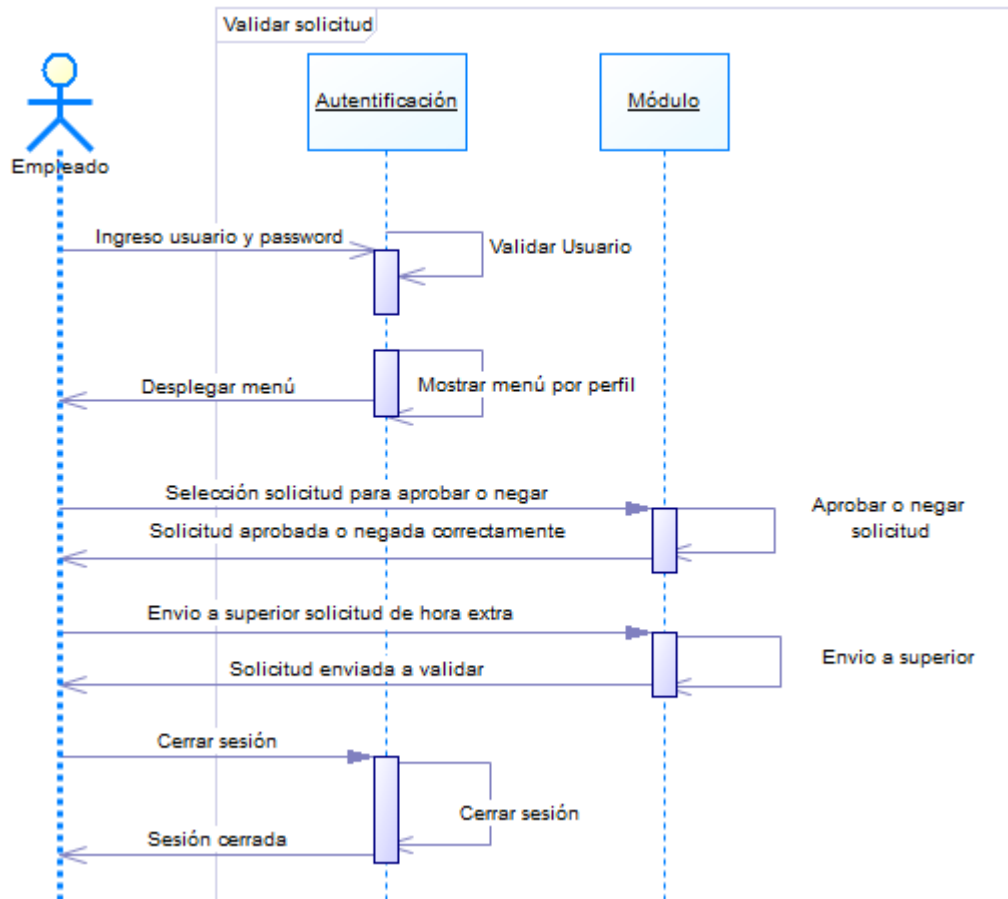


Figura 16. Diagrama de secuencia. Validación de solicitudes
Elaborado por: Paúl Hidalgo y Raúl Narváez

2.6 Modelo físico de la base de datos

En el diagrama físico de la base de datos mostrada a continuación, describe aspectos relacionados con las necesidades de la empresa para el manejo del control de asistencia del recurso humano, además que es una fuente de recopilación de datos y las relaciones entre ellos.

[illegible]

CAPÍTULO 3

CONSTRUCCIÓN Y PRUEBAS

Para la creación de este producto de software se usó el framework de desarrollo JSF 2.0, el motivo por el cual se lo uso fue porque maneja el patrón MVC (Modelo Vista Controlador), que básicamente facilitará al momento de programar, puesto que ayuda a mantener organizado el código y que a final de cuentas es de gran ayuda cuando se tiene un grupo de personas trabajando sobre el mismo proyecto y al mismo tiempo, una de las ventajas que se observó para el uso de JSF es que en la capa de presentación es decir para la vista se tiene una gran variedad de componentes con estilos únicos que realizan el producto final usado por los usuarios, el tema y layout sobre el cual se trabajó fue la librería de Sentinel 2.0, además que como objetivo principal de esta capa se muestra la interfaz para el usuario, en su mayoría están compuestos de formularios que son archivos con extensión xhtml, que a su vez son contruidos con etiquetas propias de JSF y Primefaces, otras de las herramientas que ayudó con respecto a las validaciones de dichos formularios y en la optimización de código en ciertos componentes fue Omnifaces. Con respecto a la base de datos se trabajó con PostgreSQL 9.5 y la herramienta de Mapeo de Objeto Relacional fue Hibernate que ayudó a que todos los atributos y relaciones se mapeen entre la base de datos relacional y los objetos que se están utilizando en la aplicación, es decir, la capa de persistencia de datos. Toda la lógica del negocio recae sobre los componentes desarrollados en lenguaje Java y finalmente en lo que respecta a los datos, se los accede mediante Hibernate puesto que realiza la persistencia a la base datos.

3.1 Estándares de programación

Se utiliza los estándares de programación para llevar una documentación adecuada y entendible por cada programador que forme parte del proyecto, facilitan el mantenimiento de una aplicación ya que no es muy común que el mismo programador que creó la aplicación sea el encargado de darle mantenimiento a la misma. Otro de los beneficios que se evidencia para tomar esta medida es la rápida comprensión de la lógica que se aplicó al momento de programar.

Tabla 14. Prefijos y Sufijos

Tipo de elemento	Prefijo o Sufijo	Ejemplo
Enumerados	Enum	EstadoRegistroEnum
DAO	Dao	EmpleadoDao
Servicio	Servicio	EmpleadoServicio
Controlador	Ctrl	EmpleadoCtrl
DataTable	DT	empleadoDT
CommandLink	Cmd	cmdEditarEmpleado
Form	Form	formEmpleado
PanelGrid	Png	pngEmpleado
inputText	Int	intNombreEmpleado
CommandButton	Cmd	cmdGuardarModalEmpleado

Nota: Resumen de prefijos y sufijos usados para el desarrollo del sistema
Elaborado por: Paúl Hidalgo y Raúl Narvaez

3.1.1 Notación

Una de las notaciones que ha sido usada a nivel mundial es la notación tipo camello (CamelCase) que en concepto se trata del uso de mayúsculas y minúsculas en las variables, métodos, clases, enumerados, interfaces, etc.

A continuación, se mostrará unos ejemplos de cómo se encuentra en el sistema generado dichos elementos:

Tabla 15. Notación

Identificador	Ejemplo
Clases	SegUsuario
Métodos	consultarParametrizacionByIdUsuarioAndEstado
Variables	UsuarioServicio
Constantes	REGISTRADO

Nota: Notación usada a lo largo del desarrollo del sistema

Elaborado por: Paúl Hidalgo y Raúl Narváez

Para una mejor comprensión a lo largo del desarrollo de la aplicación se coloca a cada clase con el sufijo del nombre de la capa en la que se encuentra colocada, menos en las entity, pero por ejemplo en la capa de las Dao, Servicio, Controlador, para poder identificarlos siempre van a ir con su respectivo sufijo.

3.2 Levantamiento de la aplicación web

Durante todo el desarrollo de la aplicación la mayoría de artefactos y diagramas utilizados fueron sometidos a cambios debido a la metodología de desarrollo seleccionada, ya que en las constantes reuniones que se va manteniendo por lo general saltan nuevas inquietudes y surgen cambios en algún requerimiento o a su vez aparece un nuevo requerimiento.

3.2.1 Base de datos

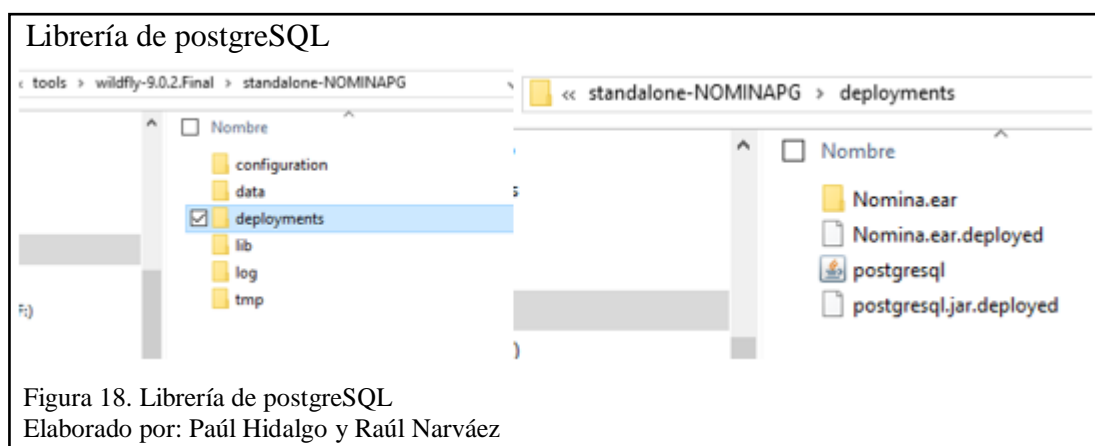
Se trata de una de las últimas etapas de la metodología del diseño de la base de datos pues como único fin, satisface los requisitos del sistema optimizando la relación que más le interesa finalmente al usuario que es la relación costo/beneficio.

Tiene un punto de inicio que es el diagrama lógico de la base de datos presentada en el anterior capítulo, cabe la aclaración que para generarlo ya se tiene escogido el gestor de base de datos que para el caso es PostgreSQL 9.5

3.2.2 Manejo de la persistencia

Una vez que se tiene definida la base de datos se procede a la inclusión de la misma con la aplicación para lo cual se usa Hibernate que viene incluido en el servidor de aplicaciones Wildfly, por esta razón se procede a la configuración de PostgreSQL sobre el servidor de aplicaciones.

La versión del Wildfly usada es la 9.0.2, uno de los primeros pasos para proceder con el procedimiento es tener el driver para PostgreSQL colocado dentro del directorio del servidor de aplicaciones en la carpeta llamada deployments para luego proceder con la configuración del datasource en Wildfly.



Dentro de la carpeta del Wildfly 9.0.2 se encuentra una carpeta llamada configuración en el que se encuentran unos archivos para la configuración que son el standalone, standalone-full, standalone-full-ha y standalone-ha. Lo que cada uno cambia es el grado de configuraciones que permite, siendo así que los archivos.

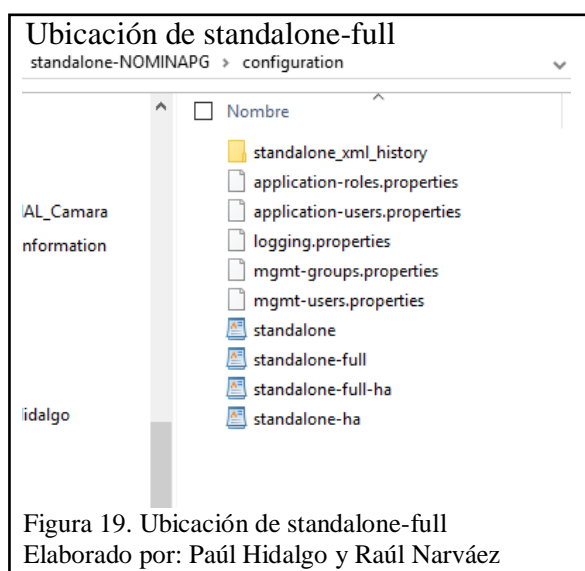
standalone.xml: es un soporte de Java EE Web-Profile más algunas extensiones como RESTful Web Services para llamadas remotas EJB3.

standalone-full.xml: Es un soporte de Java EE de perfil completo y todas las capacidades del servidor, sin agrupación.

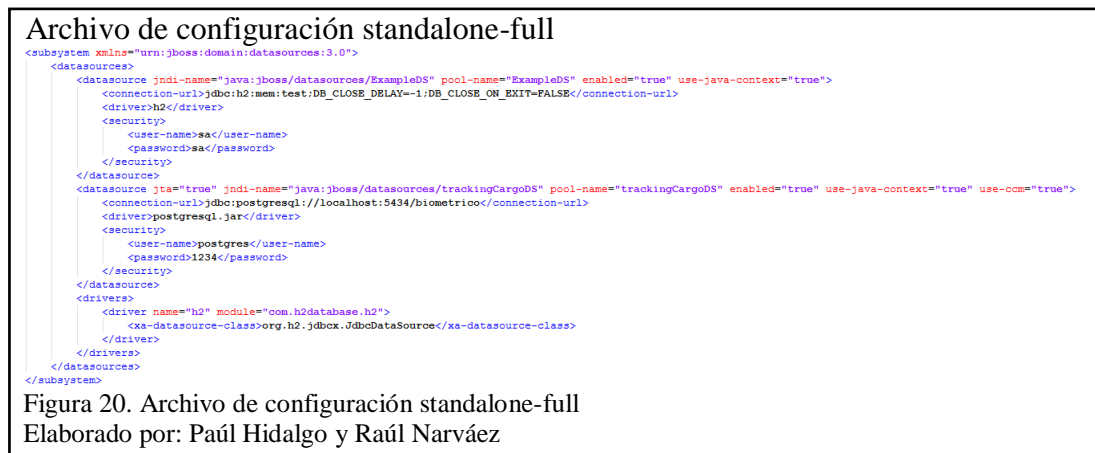
standalone-ha: Es un perfil predeterminado con capacidades de agrupación.

standalone-full-ha: Es un perfil completo con capacidades de agrupación.

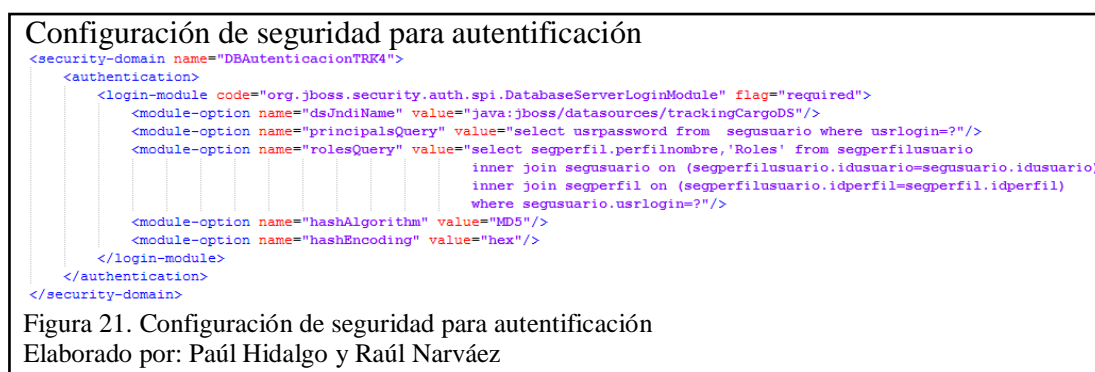
Para el caso se usará el archivo standalone-full.xml pues posee un soporte de Java EE de perfil completo y todas las capacidades del servidor sin agrupación. Una vez dentro del archivo se lo abre y busca la línea <datasource> que es donde se encuentra las configuraciones de conexión a la base de datos. A continuación, se muestra la ubicación del archivo y como se encuentra la configuración



Se procede con la configuración del datasource y del driver, para quedar como lo muestra la siguiente imagen.



Además, en el mismo archivo se configura la seguridad a cerca de aplicación que viene dentro de la etiqueta `<security-domain>` que apunta a la seguridad respecto a la autenticación consultando directamente a la base de datos y queda como lo muestra la siguiente imagen.



Lo anterior mencionado es cuanto a la configuración dentro del servidor de aplicaciones, ahora se procede con la configuración dentro de la aplicación en el archivo `persistence.xml` que para el caso se encuentra dentro de `NominaEjb/META-INF`, y se configura los parámetros que están descritos en la siguiente imagen.

Configuración de conexión con la base de datos

```
persistence.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="1.0"
3   xmlns="http://java.sun.com/xml/ns/persistence"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
6     http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">
7   <persistence-unit name="trackingCargoPU">
8     <provider>org.hibernate.ejb.HibernatePersistence</provider>
9     <jta-data-source>java:jboss/datasources/trackingCargoDS</jta-data-source>
10    <properties>
11      <property name="hibernate.show_sql" value="true" />
12      <property name="hibernate.dialect" value="org.hibernate.dialect.PostgreSQLDialect" />
13      <property name="hibernate.hbm2ddl.auto" value="update" />
14      <property name="hibernate.default_schema" value="public" />
15    </properties>
16  </persistence-unit>
17 </persistence>
```

Figura 22. Configuración de conexión con la base de datos

Elaborado por: Paúl Hidalgo y Raúl Narváez

Se generan unas clases denominadas POJOs (Plain Old Java Object) que sirven para mapear con la base de datos y permite tener una conexión directa mediante los getters y setters y que son identificadas además con anotaciones JPA.

A continuación, se mostrará la manera de cómo se encuentra conformada una entidad, resaltando las partes más importantes. Algo trascendental que se debe tener en cuenta es que no se sincroniza con la persistencia a menos que se encuentre asociada con un EntityManager, para el caso en dicha clase se encuentra configurado lo que es la unidad de persistencia y el datasource inyectado que se obtiene del contenedor EJB, además de que es genérico para tener la posibilidad de reutilizarlo y asociarlo a los diferentes tipos de entidades que existen.

Genérico para Entity Manager

```
import java.io.Serializable;
import javax.annotation.Resource;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.sql.DataSource;
import com.servitec.common.dao.impl.DaoGenericoEjb;

public class AppGenericDao<T extends Serializable, PK extends Serializable> extends DaoGenericoEjb<T, PK> {

    /**
     * Unidad de persistencia
     */
    @PersistenceContext(unitName = "trackingCargoPU")
    private EntityManager em;

    /**
     * datasource inyectado, se obtiene del contenedor EJB
     */
    @Resource(mappedName = "java:jboss/datasources/trackingCargoDS")
    private DataSource dataSource;

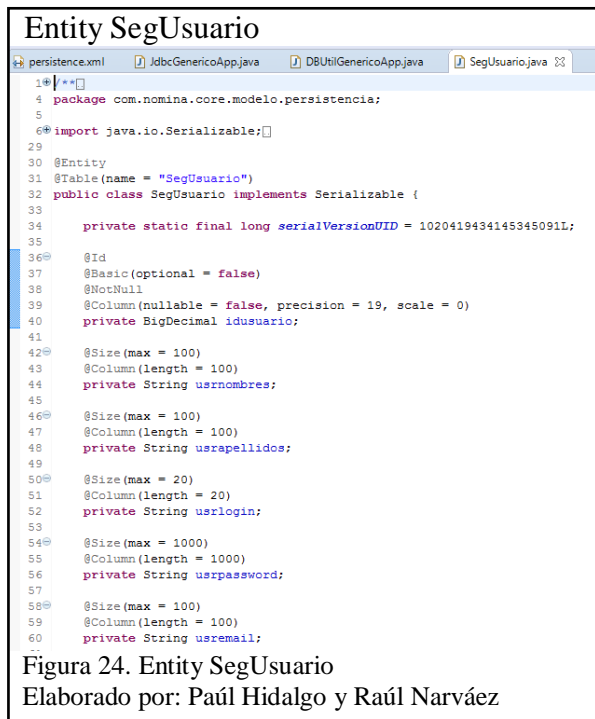
    public AppGenericDao(Class<T> type) {
        super(type);
    }

    @Override
    public DataSource getDataSource() {
        return dataSource;
    }

    @Override
    public EntityManager getEntityManager() {
        return em;
    }
}
```

Figura 23. Genérico para Entity Manager
Elaborado por: Paúl Hidalgo y Raúl Narváez

Como ejemplo se toma la clase SegUsuario que se encuentra con anotaciones JPA que la hacen una entidad y que se encuentra persistida con la base de datos, además de anotaciones que conforman parte de la Entity para no tener inconvenientes cuando se realice un deployment de la aplicación.



Para llevar a cabo los requerimientos funcionales principales es necesario obtener información de la asistencia que se carga desde un sistema biométrico externo.

En los siguientes literales se puede apreciar porciones de código que se cree relevantes.

Llena el objeto con la consulta de la hora entrada y hora salida de la asistencia. Además de retornar una validación de asistencia.

Validar hora máxima y mínima de asistencia

```
public boolean ValidaHoraMaxMinAsistencia(BigDecimal idUsuario, Date fecha) {
    boolean validaAsistencia = false;
    List<AsistenciaMinMaxQuery> asistenciaMinMaxListQuery;
    AsistenciaMinMaxQuery asistenciaMinMaxQuerySelected = new AsistenciaMinMaxQuery();
    asistenciaMinMaxListQuery = new ArrayList<>();

    try {
        asistenciaMinMaxListQuery = asistenciaServicio
            .consultarAsistenciaMaxMinByIdEmpleadoAndfechaAsistenciaAndEstado(idUsuario, fecha,
                EstadoRegistroEnum.ACTIVO);
        // Llena Clase Query con datos consultados de BD
        for (AsistenciaMinMaxQuery ammq : asistenciaMinMaxListQuery) {
            asistenciaMinMaxQuerySelected.setHoraMaxima(ammq.getHoraMaxima());
            asistenciaMinMaxQuerySelected.setHoraMinima(ammq.getHoraMinima());
        }
        // Valida que exista asistencia en fecha solicitada
        if (asistenciaMinMaxQuerySelected.getHoraMaxima() == null
            || asistenciaMinMaxQuerySelected.getHoraMinima() == null) {
            validaAsistencia = false;
        } else {
            validaAsistencia = true;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return validaAsistencia;
}
```

Figura 25 Validar hora máxima y mínima de asistencia
Elaborado por: Paúl Hidalgo y Raúl Narváez

Código relevante sobre el ingreso de horario por plan de trabajo

```
if (asistenciaMinMaxQuerySelected.getHoraMaxima() != null
    && asistenciaMinMaxQuerySelected.getHoraMinima() != null) {

    // Llena horario por plan
    horarioTrabajoQuerySelected = consultarHorasEntradaSalidaHorarioPlanByIdUsuarioAndFechaActividad(
        idUsuario, fecha);

    // Sumatoria de tiempo de actividades de fecha seleccionada
    sumatoriaRegistroActividadesQuerySelected = consultarSumatoriaRegistroActividadByIdUsuarioAndFechaActividad(
        idUsuario, fecha, EstadoRegistroEnum.ACTIVO);

    // Llenar variables asistencia para hacer operaciones
    long horaEntradaAsistencia = asistenciaMinMaxQuerySelected.getHoraMinima().getTime();
    long horaSalidaAsistencia = asistenciaMinMaxQuerySelected.getHoraMaxima().getTime();

    // Valida si existe fecha en Horario segun plan de trabajo
    if (horarioTrabajoQuerySelected != null) {
        // Llamar a método que hace el cálculo con horario Plan
        horaExtra = calculoHoraOfHorarioPlan(horarioTrabajoQuerySelected.getHoraInicioHorarioPlan(),
            horarioTrabajoQuerySelected.getHoraFinHorarioPlan(), horaEntradaAsistencia,
            horaSalidaAsistencia, sumatoriaRegistroActividadesQuerySelected.getSumaTiempoActividades());
        if (horaExtra <= 0) {
            horaExtra = 0;
        }
    } else {
        // Llena horario por defecto
        horarioTrabajoQuerySelected = consultarHorasEntradaSalidaHorarioDefaultByIdUsuarioAndFechaActividad(
            idUsuario, fecha);
        // Llamar a método que hace el cálculo con horario default
        horaExtra = calculoHoraOfHorarioDefault(horarioTrabajoQuerySelected.getHoraInicioHorarioDefault(),
            horarioTrabajoQuerySelected.getHoraFinHorarioDefault(), horaEntradaAsistencia,
            horaSalidaAsistencia, sumatoriaRegistroActividadesQuerySelected.getSumaTiempoActividades());
        if (horaExtra <= 0) {
            horaExtra = 0;
        }
    }
}
```

Figura 26 Código relevante sobre el ingreso de horario por plan de trabajo
Elaborado por: Paúl Hidalgo y Raúl Narváez

Cálculo de hora extra para horario por defecto

```
// Cálculo de hora extra para horario Default
public long calculoHoraOfHorarioDefault(Time horaInicioHorarioDefault, Time horaFinHorarioDefault,
    long horaEntradaAsistencia, long horaSalidaAsistencia, long sumatoriaRegistroActividades) {

    // Llenar variables plan para hacer operaciones
    long horaEntradaDefault = horaInicioHorarioDefault.getTime();
    long horaSalidaDefault = horaFinHorarioDefault.getTime();

    horaEntradaAsistencia = CalcularMinutosGracia(horaEntradaDefault, horaEntradaAsistencia);
    horaSalidaAsistencia = CalcularMinutosGracia(horaSalidaDefault, horaSalidaAsistencia);

    // Cálculo de hora extra diferencia de entrada mas
    // diferencia de salida menos tiempo de actividades
    // registradas esa fecha
    long diferenciaEntrada = ((horaEntradaDefault - horaEntradaAsistencia) / 60000);
    long diferenciaSalida = ((horaSalidaAsistencia - horaSalidaDefault) / 60000);
    long tiempoActividadesRegistradas = (sumatoriaRegistroActividades) / 60000;

    long horaExtra = (diferenciaEntrada + diferenciaSalida) - tiempoActividadesRegistradas;

    return horaExtra;
}
```

Figura 27. Cálculo de hora extra para horario por defecto
Elaborado por: Paúl Hidalgo y Raúl Narváez

Cálculo de hora extras para horario por plan

```
// Cálculo de hora extra para horario Plan
public long calculoHoraOfHorarioPlan(Time horaInicioHorarioPlan, Time horaFinHorarioPlan,
    long horaEntradaAsistencia, long horaSalidaAsistencia, long sumatoriaRegistroActividades) {

    // Llenar variables plan para hacer operaciones
    long horaEntradaPlan = horaInicioHorarioPlan.getTime();
    long horaSalidaPlan = horaFinHorarioPlan.getTime();

    horaEntradaAsistencia = CalcularMinutosGracia(horaEntradaPlan, horaEntradaAsistencia);
    horaSalidaAsistencia = CalcularMinutosGracia(horaSalidaPlan, horaSalidaAsistencia);

    // Cálculo de hora extra diferencia de entrada mas
    // diferencia de salida menos tiempo de actividades
    // registradas esa fecha
    long diferenciaEntrada = ((horaEntradaPlan - horaEntradaAsistencia) / 60000);
    long diferenciaSalida = ((horaSalidaAsistencia - horaSalidaPlan) / 60000);
    long tiempoActividadesRegistradas = (sumatoriaRegistroActividades) / 60000;

    long horaExtra = (diferenciaEntrada + diferenciaSalida) - tiempoActividadesRegistradas;

    return horaExtra;
}
```

Figura 28. Cálculo de hora extras para horario por plan
Elaborado por: Paúl Hidalgo y Raúl Narváez

Cálculo de minutos de gracia

```
public long CalcularMinutosGracia(long horaHorario, long horaAsistencia) {  
  
    try {  
        // Consulta dato de tabla parámetros genéricos  
        long minutosDeGracia = parametrosGenericosServicio.consultarParametro(PGMenuEnum.MINUTOS_DE_GRACIA,  
            TipoRetornoParametroGenerico.LONG);  
  
        long minutosAsistencia = (horaAsistencia / 60000);  
        long minutosHorarioMenosGracia = (horaHorario / 60000) - minutosDeGracia;  
        long minutosHorarioMasGracia = (horaHorario / 60000) + minutosDeGracia;  
  
        // validación para considerar hora extra o atraso  
        if (minutosAsistencia >= minutosHorarioMenosGracia && minutosAsistencia <= minutosHorarioMasGracia) {  
            horaAsistencia = horaHorario;  
        }  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
    return horaAsistencia;  
}
```

Figura 29. Cálculo de minutos de gracia

Elaborado por: Paúl Hidalgo y Raúl Narváez

3.3 Plan de pruebas

El propósito del plan de pruebas es proveer la información necesaria para establecer las técnicas, herramientas y actividades relacionadas con la ejecución y validación de cada una de las pruebas de aceptación del sistema a realizar, permite planear y controlar los esfuerzos de pruebas del proyecto o de las iteraciones específicas, incluye responsabilidades de cada una de las tareas, los recursos y los prerequisites que deben ser considerados en el esfuerzo de cada una de las pruebas, permitiendo garantizar el cumplimiento de los requerimientos planteados.

El paso posterior al desarrollo de un sistema de software son los diferentes tipos de pruebas que debe pasar el sistema, para ello se crea un plan de pruebas el cual nos ayuda a identificar las funcionalidades a ser probados y los recursos usados para todo el proceso de pruebas de sistema.

El plan de pruebas es un instrumento muy útil para realizar pruebas ya que ayuda a asegurar que el software hace o que debería hacer además de identificar problemas que se los pueda corregir antes de que un sistema sea usado por el usuario.

El plan de pruebas fue realizado usando los casos de uso y funcionalidades del producto final.

3.3.1 Pruebas funcionales

Las pruebas funcionales se consideran como un conjunto de actividades que se realizan para comprobar determinados aspectos de un sistema de software además de que gracias a estas se verifica el comportamiento del sistema, subsistema o componente de software descrito en especificaciones de requerimientos o casos de uso.

Para el desarrollo de pruebas funcionales no se utiliza ninguna herramienta extra en especial debido a que constituye probar la aplicación en base a los casos de uso.

Actividades para realizar pruebas funcionales

Generar casos de prueba.

Validación de casos de prueba.

Registro de defectos encontrados en el sistema.

3.3.2 Pruebas no funcionales

Las pruebas no funcionales están en base a los requisitos no funcionales los cuales por su definición no describen información a guardar, ni funciones a realizar sino características de funcionamiento por ejemplo se puede considerar las pruebas de disponibilidad, rendimiento, y usabilidad.

Para la ejecución de pruebas no funcionales es muy recomendable usar herramientas externas las cuales ayudan a ejecutar estas pruebas con mayor facilidad. En el presente proyecto se usa la herramienta JMETER para realizar dichas pruebas.

Funcionalidades a probar.

En el presente fragmento de texto se lista las funcionalidades generales de los diferentes módulos del sistema desarrollado.

Módulo Seguridades: Administración de módulos, administración de Perfiles, administración de tipo de opción, administración de permiso, administración de opciones.

Módulos Nomina.

Administración: Gestión de horario, planificación de festividades, administración de empleados.

Registro de asistencia: Registro manual, carga de información.

Actividades personales: Registro de actividades, crear nueva solicitud.

Validar solicitud: Aprobar/Negar solicitudes.

Reportes: Reporte personal, reporte administrativo, reporte Gerencial.

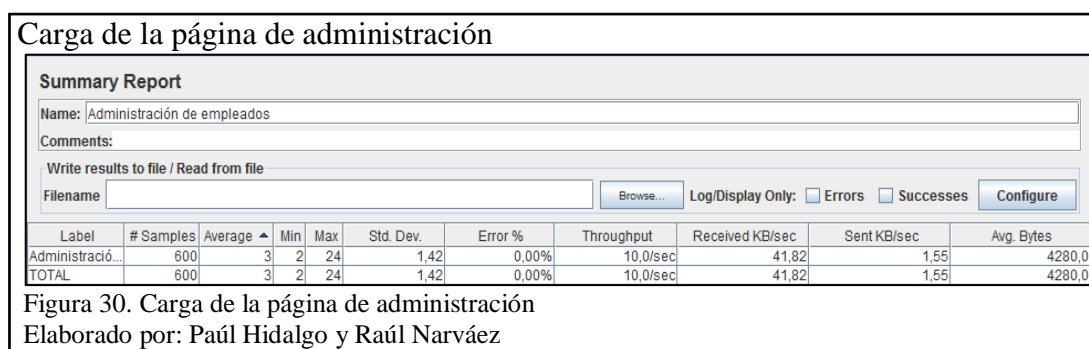
Además, se enfocan en certificar el sistema por medio de sus requerimientos y el funcionamiento general del software, es decir la forma en que el sistema funciona y no por el comportamiento de funciones específicas. Para ello se califica el tiempo de respuesta de la aplicación a solicitudes en un ambiente crítico.

Para ello usamos la herramienta JMeter el cual es una herramienta para analizar y medir desempeño de una variedad de servicios. Entre ellos lo que vamos a destacar son pruebas de carga y de estrés.

3.3.3 Pruebas de rendimiento

Son ejecutadas mediante una simulación que facilita la herramienta en donde se especifica unos grupos de actores ejecutando las pruebas según los perfiles credos en la aplicación. Para ello tomamos los siguientes procesos: Administración de empleados, Administración de solicitudes y Generación de reportes. Para someterse a las pruebas tomamos 600 solicitudes de usuarios y hacemos que las solicite esperando que el sistema responda de la mejor forma.

En la presente matriz que brinda JMeter se puede ver los datos principales sobre la carga de página de Administración de empleados una de las más concurridas por los usuarios, Se ejecutó para 100, 200 y 600 usuarios en 60 segundos.



En la siguiente matriz se puede observar la carga de la página Crear solicitud la cual se puede notar que el margen de Error es 0% lo cual es positivo sobre las respuestas a solicitudes de usuarios concurrentes de la misma forma esto se ejecutó para 100, 200 y 600 usuarios en 60 segundos.

Carga de la página al crear solicitud

Summary Report

Name: Administración de solicitudes

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
Administració...	600	2	1	30	1,58	0,00%	10,0/sec	41,83	1,54	4280,0
TOTAL	600	2	1	30	1,58	0,00%	10,0/sec	41,83	1,54	4280,0

Figura 31. Carga de la página al crear solicitud

Elaborado por: Paúl Hidalgo y Raúl Narváez

Otra de las matrices que muestran un rendimiento positivo ante la carga es la de exportación de reportes teniendo como tiempo de respuesta promedio de 1.3 /sec un tiempo muy bueno ante una carga de 600 usuarios en 60 segundos.

Exportación de reportes

Summary Report

Name: Exportar reportes

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
Exportar repo...	1220	2	1	13	0,96	0,00%	1,3/sec	5,40	0,21	4280,0
TOTAL	1220	2	1	13	0,96	0,00%	1,3/sec	5,40	0,21	4280,0

Figura 32. Exportación de reportes

Elaborado por: Paúl Hidalgo y Raúl Narváez

Dentro de los requerimientos no funcionales es muy importante medir la facilidad de uso, entendimiento, efectividad y eficiencia para ello se hace una calificación cuantitativa sobre facilidades que brinde el sistema para que el usuario pueda tener una guía sobre el funcionamiento.

Tabla 16. Criterios a calificar de requerimientos no funcionales

Criterio	Calificación
Ayuda navegacional.	4
Ayuda y documentos en línea.	5
Teclas de funciones pre asignadas.	4
Utilizaciones de zonas destacadas, subrayados, coloreados, y otros	3

indicadores que ayuden al usuario a localizar elementos	
El menor número de pantallas posible para implementar la funcionalidad	5

Nota: Descripción de los criterios a calificar de requerimientos no funcionales

Elaborado por: Paúl Hidalgo y Raúl Narváez

Tomando en cuenta la siguiente tabla de calificaciones.

Tabla 17. Criterios de calificación

Malo	Regular	Bueno	Muy bueno	Excelente
1	2	3	4	5

Nota: Criterios de calificación para pruebas no funcionales

Elaborado por: Paúl Hidalgo y Raúl Narváez

3.3.3 Ambiente de pruebas

Para superar las pruebas funcionales y no funcionales se tendrán datos reales si se elabora un ambiente de pruebas con características mínimas.

Para ello se utilizará una maquina con características mínimas que sirva como servidor de aplicaciones y de la misma forma como servidor de base de datos.

Tabla 18. Herramientas usadas para pruebas

Equipo	Procesador	Memoria	Disco duro	Sistema operativo	Pantalla
Xtratech	Core i3-3110M	4GB	300GB	Windows 8.1	15 pulgadas

Nota: Herramientas usadas para instalar ambiente de pruebas

Elaborado por: Paúl Hidalgo y Raúl Narváez

Tabla 19. Plan de pruebas.

Módulo	Actor	Pre Requisito	Proceso	Pasos	Resultado esperado	Resultado alcanzado
Administración de seguridades del sistema	Administrado de software	Acceso a sistema mediante usuario y contraseña.	Administración de módulos	1. Ingresar nombre	Guardar y visualizar módulo ingresado	OK
				2. Ingresar descripción		
				3. Ingresar iniciales		
				4. Ingresar URL.		
				5. Ingresar porcentaje de desarrollo		
				6. Seleccionar ícono		
				7. Seleccionar color de ícono		
				8. Seleccionar tipo de módulo		
				9. Seleccionar estado		
			Administración de perfil	1. Ingresar nombre	Guardar y visualizar perfil ingresado	OK
				2. Ingresar descripción		
				3. seleccionar estado		
			Administración de tipo de opción	1. Ingresar nombre	Guardar y visualizar tipo de opción	OK
				2. Ingresar descripción		
				3. Seleccionar estado		
			Administración de permisos	1. Seleccionar perfil	Guardar y visualizar usuarios asignados a perfil	OK
				2. Seleccionar usuarios		
				3. Ingresar descripción		
				4. Seleccionar estado		
			Administración de opciones	1. Seleccionar módulo	Guardar y visualizar opción ingresada.	OK
				2. Seleccionar tipo de opción		
				3. Ingresar nombre		
				4. Ingresar URL.		
				5. Ingresar descripción		

				6. Seleccionar ícono		
				7. Seleccionar estado.		
			Administración de perfiles a opciones	1. Seleccionar opción	Guardar y visualizar asignación de perfil a opción	OK
				2. Seleccionar perfil		
				3. Ingresar Observaciones		
				4. Seleccionar estado		

Nota: Descripción del plan de pruebas

Elaborado por: Paúl Hidalgo y Raúl Narváez

CONCLUSIONES

- Con la implementación de aplicación se facilita el control y gestión de la asistencia del recurso humano de la empresa, debido a que los procesos de control y de gestión están automatizados en el mismo
- El uso de una metodología de desarrollo ágil ayuda al desarrollo de la aplicación, ya que prioriza el desarrollo del software, la interacción con el dueño del producto, y facilita la gestión de los cambios en los requerimientos
- Para desarrollar un proyecto de este tipo, se requiere que los integrantes del equipo de desarrollo tengan un conocimiento claro de la plataforma de desarrollo seleccionada para desarrollar el proyecto.

RECOMENDACIONES

- Para el desarrollo de un sistema de gestión de la información relacionada con el control de asistencia de recursos humanos de una empresa es necesario estudiar los artículos oficiales gubernamentales relacionados con el código de trabajo. Además de que nos ayuda como guía para registrar posibles requerimientos que presenta una entidad que maneje un número considerable de empleados.
- Para garantizar la calidad del producto de software es necesario definir un plan de pruebas para la evaluación del sistema que se basa en los requerimientos funcionales y no funcionales del sistema (pruebas de caja blanca, pruebas de caja negra, pruebas de carga).

LISTA DE REFERENCIAS

- Albaladejo, X. (2008). *Proyectos Agiles*. Obtenido de <https://proyectosagiles.org/que-es-scrum/>
- Ceria, S. (02 de 2001). *Departamento de Computación*. Recuperado el Mayo de 2017, de http://www-2.dc.uba.ar/materias/isoft1/2001_2/apuntes/CasosDeUso.pdf
- Git. (2017). *Git*. Obtenido de <https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones>
- Gómez Estupiñan, J. F. (08 de 04 de 2014). *BIBLIOTECA, UNIVERSIDAD DE MANIZALES*. Obtenido de <http://revistasum.umanizales.edu.co/ojs/index.php/ventanainformatica/article/viewFile/274/397>
- Rumbaugh, J., Jacobson, I., & Booch, G. (2000). *El Lenguaje Unificado de Modelado. Manual de Referencia*. Madrid, España: Addison Wesley. Obtenido de <http://www.face.ubiobio.cl/~cvidal/modelamiento/libros/LenguajeUnificadoModelado.pdf>
- Salinas Caro, P., & Hitschfeld Kahler, N. (s.f.). *Departamento de Ciencias de la Computación, Universidad de Chile*. Recuperado el 06 de 2017, de <https://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>
- Sistemas, L. (2013). *Leal Sistemas*. Obtenido de <http://www.lealsistemas.com.ar/relojes-de-control.html>
- Systems, S. (2017). *SPARK SYSTEMS*. Recuperado el 06 de 2017, de http://www.sparxsystems.com.ar/resources/tutorial/uml2_sequencediagram.html
- tecnologias-informacion*. (2016). Obtenido de <http://www.tecnologias-informacion.com/gestionrh.html>
- Torres, L. (12 de 10 de 2007). *Gravitar*. Obtenido de <http://gravitar.biz/bi/metodologias-agiles-intro/>