



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS  
ELECTRÓNICA E INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS  
COMPUTACIONALES E INFORMÁTICOS**

**Tema:**

---

EMPLEO DE TECNOLOGIA GWT EN LA CREACION DE UN  
SISTEMA WEB DE CONTROL DE ASISTENCIA APLICANDO  
BIOMETRIA DACTILAR PARA EL PERSONAL DE LA FABRICA  
TEXTIL GMB UBICADA EN EL CANTON SAN PEDRO DE PELILEO

---

Proyecto de Titulación o Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

Sublínea de Investigación: Aplicaciones web

AUTOR: Wilson Iván Sánchez Paredes

TUTOR: Ing. Mg. Franklin Mayorga

Ambato - Ecuador

Octubre/2017

## APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el tema: **“Empleo de tecnología GWT en la creación de un sistema web de control de asistencia aplicando biometría dactilar para el personal de la fábrica textil GMB ubicada en el cantón San Pedro de Pelileo”**, del señor Sánchez Paredes Wilson Iván , estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato octubre, 2017

EL TUTOR



---

Ing. Mg. Franklin Mayorga

## AUTORÍA

El presente Proyecto de Investigación titulado: **“Empleo de tecnología GWT en la creación de un sistema web de control de asistencia aplicando biometría dactilar para el personal de la fábrica textil GMB ubicada en el cantón San Pedro de Pelileo”**, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato octubre, 2017



Sánchez Paredes Wilson Iván  
CC: 1805042932

## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato octubre, 2017



Sánchez Paredes Wilson Iván  
CC: 1805042932

## APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes: Ing. Mg. Carlos Nuñez, Ing Oswaldo Paredes , revisó y aprobó el Informe Final del Proyecto de Investigación titulado **“Empleo de tecnología GWT en la creación de un sistema web de control de asistencia aplicando biometría dactilar para el personal de la fábrica textil GMB ubicada en el Cantón San Pedro de Pelileo”**, presentado por el señor Sánchez Paredes Wilson Iván de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ing. Mg. Pilar Urrutia  
PRESIDENTE DEL TRIBUNAL

Ing. Mg. Oswaldo Paredes  
DOCENTE CALIFICADOR

Ing. Mg. Carlos Nuñez  
DOCENTE CALIFICADOR

## **DEDICATORIA:**

A Dios por darme la salud, fuerza y conocimiento para seguir adelante con mi vida profesional.

A mi madre que es fuente de inagotable de lucha, coraje y un gran ejemplo de perseverancia.

A mi hermano que desde el cielo me guía y apoya en mis momentos de dificultad.

A mi padre y hermanos que siempre me estuvieron apoyando, tanto económicamente como emocionalmente, durante el transcurso de la carrera.

A mis amigos y compañeros que supieron brindarme su apoyo durante el transcurso de aprendizaje.

Sánchez Paredes Wilson Iván

## **AGRADECIMIENTO:**

Agradezco a todo el personal tanto docente como administrativo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial por brindarme sus conocimientos y darme la oportunidad de haberme formado profesionalmente.

Un sincero agradecimiento al Ing. Mg. Franklin Mayorga por haberme guiado en todo este proceso de elaboración de tesis impartíendome sus conocimientos.

Sánchez Paredes Wilson Iván

## INDICE

INDICE .....	viii
CAPITULO 1 .....	16
EL PROBLEMA .....	16
1.1. Tema.....	16
1.2. Planteamiento del problema.....	16
1.3. Delimitación.....	19
1.4. Justificación.....	19
1.5. Objetivos .....	22
1.5.1. Objetivo General .....	22
1.5.2. Objetivos Específicos .....	22
CAPITULO 2.....	23
MARCO TEORICO .....	23
2.1. Antecedentes Investigativos .....	23
2.2. Fundamentación Teórica.....	26
2.2.1. Lenguaje Java .....	26
2.2.2. JDK.....	26
2.2.3. Desarrollo de Software .....	26



2.2.4.	Tecnología WEB .....	27
2.2.5.	Desarrollo Web en Java.....	27
2.2.6.	Arquitectura WEB.....	27
2.2.7.	IDE de desarrollo.....	30
2.2.8.	Ajax .....	30
2.2.9.	Bases de Datos.....	30
2.2.10.	RIAs.....	31
2.2.11.	Google Web Toolkit .....	32
2.2.12.	Características de GWT:.....	32
2.2.13.	Arquitectura GWT .....	33
2.2.14.	Biometría .....	34
2.2.15.	Biometría Dactilar.....	34
2.2.16.	Propuesta de Solución.....	36
CAPITULO 3.....		37
METODOLOGIA .....		37
3.1.	Modalidad de la Investigación .....	37
3.2.	Recolección de Información .....	37
3.3.	Procesamiento y Análisis de Datos.....	38
3.4.	Desarrollo del Proyecto .....	38
CAPITULO 4.....		40

DESARROLLO DE LA PROPUESTA .....	40
4.1. Análisis del Sistema y Requerimientos .....	40
4.1.1. Recolección y Análisis de Información .....	40
4.1.2. Levantamiento de los Requerimientos del Sistema .....	43
4.1.3. Selección de Herramientas para el Desarrollo de la Aplicación Web .....	44
4.1.4. Definición de los Requerimientos.....	48
4.2. Diseño de Esquemas de la Aplicación Web .....	51
4.2.1. Diagramas Casos de Uso .....	52
4.2.1.1. Descripción de los Actores .....	54
4.2.1.2. Descripción de los Casos de Uso .....	55
4.2.1.3. Diagramas de Secuencia .....	57
4.2.2. Diseño de Base de Datos .....	61
4.2.2.1. Diccionario de Datos .....	62
4.2.3. Diseño de la Interfaz de Usuario.....	65
4.3. Codificación de la Aplicación Web .....	66
4.3.1. Codificación del Lado del Cliente .....	66
4.3.2. Codificación del Lado del Servidor .....	83
4.4. Pruebas de funcionamiento.....	88
4.5. Implantación de la Aplicación Web.....	95
4.6. Mantenimiento de la Aplicacion Web .....	96

CAPITULO V: CONCLUSIONES Y RECOMENDACIONES.....	99
5.1. Conclusiones.....	99
5.2. Recomendaciones .....	100
ANEXOS .....	105

## INDICE DE TABLAS

Tabla 1. Descripción del Actor Colaborador. Elaborado por: El Investigador.....	54
Tabla 2. Descripción del Actor Secretaria. Elaborado por: El Investigador.....	54
Tabla 3. Descripción Registrar hora de ingreso a jornada laboral. Elaborado por: El Investigador .....	55
Tabla 4. Descripción de Registrar hora de salida de jornada laboral. Elaborado por: El Investigador .....	55
Tabla 5. Descripción de Registrar anomalías en la asistencia. Elaborado por: El Investigador .....	56
Tabla 6. Descripción de Generar informe de asistencia. Elaborado por: El Investigador	56
Tabla 7. Descripción de Generar reporte de días laborados por mes. Elaborado por: El Investigador .....	56
Tabla 8. Descripción de Generar reporte de asistencia por mes. Elaborado por: El Investigador .....	57
Tabla 9. Diccionario de datos de la tabla “Atrasos”. Elaborado por el investigador.....	62
Tabla 10. Diccionario de datos de la tabla “Colaboradores”. Elaborado por el investigador.....	62
Tabla 11. Diccionario de datos de la tabla “Cargos”. Elaborado por el investigador .....	63
Tabla 12. Diccionario de datos de la tabla “Faltas”. Elaborado por el investigador .....	63
Tabla 13. Diccionario de datos de la tabla “Jornadas”. Elaborado por el investigador....	63
Tabla 14. Diccionario de datos de la tabla “Jornadas_Periodos”. Elaborado por el investigador.....	63

Tabla 15.Diccionario de datos de la tabla “Periodos”. Elaborado por el investigador....	64
Tabla 16.Diccionario de datos de la tabla “Registros”. Elaborado por el investigador...	64
Tabla 17.Diccionario de datos de la tabla “Usuarios”. Elaborado por el investigador....	64
Tabla 18. Cronograma de implantación de la aplicación web .....	95

## ÍNDICE DE IMÁGENES

Imagen 1. Arquitectura GWT. Elaborado por: El Investigador.....	34
Imagen 2. Huellas Dactilares. Elaborado por: <a href="http://www.biometria.gov.ar/metodos-biometricos/dactilar.aspx">http://www.biometria.gov.ar/metodos-biometricos/dactilar.aspx</a> .....	35
Imagen 3.Diagrama de Proceso. Elaborado por: El Investigador .....	43
Imagen 4. Estructura del Sistema. Elaborado por: El Investigador .....	44
Imagen 5. Controles de Formulario HTML 5. Elaborado por: <a href="http://desarrolloweb.dlsi.ua.es/">http://desarrolloweb.dlsi.ua.es/</a> .....	49
Imagen 6. Atributos de Formulario HTML5. Elaborado por: <a href="http://desarrolloweb.dlsi.ua.es/">http://desarrolloweb.dlsi.ua.es/</a> .....	50
Imagen 7.Casos de Uso. Elaborado: Por el Investigador.....	53
Imagen 8. Caso de Uso Colaborador. Elaborado por el Investigador.....	53
Imagen 9. Caso de Uso Secretaria. Elaborado por: El Investigador .....	54
Imagen 10.Diagrama de secuencia de registro de ingresos. Elaborado por el investigador.....	58
Imagen 11. Diagrama de secuencia de registro de salida. Elaborado por el investigador. ....	58
Imagen 12. Diagrama de secuencia de Validar registros de asistencia. Elaborado por el investigador.....	59
Imagen 13. Diagrama de secuencia de generación de informes de asistencia. Elaborado por el investigador. ....	59

Imagen 14. Diagrama de secuencia de Generar reporte de días laborados en el mes. Elaborado por el investigador .....	60
Imagen 15. Diagrama de secuencia de Generar reporte de asistencia mensual. Elaborado por el Investigador .....	60
Imagen 16. Diseño Base de Datos. Elaborado por: El Investigador.....	61
Imagen 17. Interfaz de login. Elaborado por el investigadot.....	65
Imagen 18. Interfaz de la página administradora de las funcionalidades del software. Elaborado por el investigador.....	65
Imagen 19. Estructura genérica de las interfaces para manipular la información de la base de datos. Elaborado por el Investigador. ....	66
Imagen 20. Login de la aplicación web. Elaborado por el autor .....	89
Imagen 21. Página principal de la aplicación web. Elaborado por el investigador .....	89
Imagen 22. Página de registro de colaboradores. Elaborado por el investigador .....	90
Imagen 23. Página de registro de asistencia. Elaborado por el investigador .....	90
Imagen 24. Reporte de asistencia mensual. Elaborado por el investigador.....	91

# CAPITULO 1

## EL PROBLEMA

### 1.1. Tema

“Empleo de tecnología GWT en la creación de un sistema web de control de asistencia aplicando biometría dactilar para el personal de la fábrica textil GMB ubicada en el cantón San Pedro de Pelileo”.

### 1.2. Planteamiento del problema

Actualmente vivimos en un mundo de constante cambio y crecientes necesidades, donde las aplicaciones web han evolucionado mucho, y ya es habitual que en las especificaciones se exijan características que no encajan con el paradigma web guardando más similitud a las aplicaciones desarrolladas para escritorio. Es por ello que ha surgido un nuevo término, el de aplicaciones ricas en Internet, éstas son aplicaciones web pero con una funcionalidad similar a las aplicaciones de escritorio, mejorando así considerablemente la experiencia final de usuario, estas se ejecutan en un navegador estandarizado, considerándole a éste el sistema operativo anfitrión. En las aplicaciones web tradicionales se produce una carga continua de páginas y recursos por lo que se produce un tráfico muy elevado entre cliente y servidor; en una aplicación rica en internet (RIAs) esto no sucede por dos razones, la primera es que la aplicación se carga completamente desde el principio, y la segunda es que la comunicación que las RIA establecen con el servidor será asíncrona y mediante el uso de AJAX.

Google Web Toolkit (GWT) es un conjunto de herramientas de desarrollo web pensado para construir y optimizar RIAs. Su objetivo es abstraer al programador de las



dificultades propias de la programación web, tales como el difícil manejo de excepciones, depuración deficiente o nula, y por supuesto de las peculiaridades de cada navegador.

Cabe destacar también que GWT es utilizado por muchos productos de Google, incluyendo Google Wave y la nueva versión de AdWords. GWT es de código abierto, totalmente gratuito, y utilizado por miles de desarrolladores en todo el mundo. Desarrollar aplicaciones web, teniendo en cuenta cada una de las diferencias entre los navegadores web más populares en el mercado, puede llegar a ser muy tedioso y propenso a errores; en ocasiones se puede llegar a emplear el 90% del tiempo del desarrollo a resolver las peculiaridades propias de cada navegador, en vez de dedicarlo a realizar una buena arquitectura. [1]

Tomando en cuenta las ventajas y beneficios que la tecnología GWT ofrece se pretende aplicar dicha tecnología para cubrir una necesidad encontrada en la fábrica textil GMB ubicada en el Cantón Pelileo, Provincia de Tungurahua en el Km 3 vía a Huambaló.

Actualmente todas las empresas, industrias, fábricas o instituciones con o sin fines de lucro necesitan de la informática para automatizar y mejorar sus procesos manuales permitiendo a las organizaciones ser más eficaces y competitivas en un mundo donde los avances de la tecnología de la información vuelve al mercado más hostil para las empresas que no explotan o no invierten en dicha tecnología para ser más robustas y competitivas en el mercado actual.

El cantón Pelileo o también conocido como la ciudad azul, donde la mayor parte de sus habitantes se dedican a la producción de prendas textiles en su mayoría de jean aportan al desarrollo económico del cantón, al ser esta la principal actividad comercial los índices de competitividad entre las fábricas es muy alto, obligando a los propietarios de las mismas a buscar nuevas formas de obtener ventajas competitivas del mercado en el que se desenvuelven.

La fábrica de prendas jean “GMB” ubicada en el cantón Pelileo, Km 3 vía a Huambaló actualmente realiza el proceso de control de asistencia y cumplimiento de jornadas laborales del personal de forma manual, provocando inconsistencia en los reportes diarios.

Las áreas de trabajo carecen de un control adecuado del cumplimiento de la jornada laboral como consecuencia del manejo obsoleto de los registros de asistencia, predominando la suplantación de identidad, impuntualidad y la ausencia del personal que conforman la fábrica.

La fábrica ha tenido un notable declive en la producción semanal programada de prendas jean a consecuencia del ausentismo y retraso al área de trabajo provocando pérdidas económicas graves, disminución de las remuneraciones y pérdida de credibilidad de la fábrica frente a la competencia, esta situación problemática se agrava en las temporadas donde crece la demanda de producción y es necesario optimizar y controlar las jornadas laborales para satisfacer los niveles de utilidad esperados para dichas temporadas.

El incumplimiento del plan de producción semanal no solo ocasiona pérdidas económicas a corto plazo sino que repercute notablemente en los balances de utilidades que se generan al final de los periodos fiscales, obligando a la gerencia a extender las jornadas laborales para satisfacer la demanda actual del mercado.

La extensión de las jornadas laborales no solo supone un esfuerzo extra por parte del personal, también representa un mayor uso de los recursos que requiere la fábrica para funcionar, un mayor desgaste de la maquinaria, un consumo excesivo de energía eléctrica, agua potable, aire acondicionado y demás, lo que acarrea recargos económicos en las planillas de consumo de los servicios básicos y mantenimiento de la maquinaria.

Basado en los antecedentes citados, surge la necesidad de implementar tecnología GWT en la creación de un sistema web de control que asegure que el personal que conforma la fábrica textil GMB cumpla con las jornadas laborales dentro de los horarios establecidos en la misma.

### **1.3. Delimitación**

**Área Académica:** Software

**Línea de Investigación:** Desarrollo de Software

**Sub líneas de Investigación:** Aplicación Web

#### **Delimitación Espacial:**

El presente proyecto se desarrollará en la fábrica de prendas jean “GMB” ubicado en el cantón Pelileo, Provincia de Tungurahua Km 3 vía a Huambaló.

#### **Delimitación Temporal:**

El tiempo estimado del proyecto será durante el periodo Abril 2016 – Septiembre 2016.

### **1.4. Justificación**

Es importante destacar que cada vez la tecnología de la información evoluciona a pasos agigantados, lo que se ve reflejada en la aparición de herramientas que cubren las constantes necesidades de las empresas y aportan directamente al desarrollo económico y expansión de las mismas, la base de la tecnología de la información es la comunicación la misma que en la actualidad se ha orientado hacia el internet que se ha convertido en el principal instrumento de comunicación en el mundo, además de permitir transmitir nuestras ideas y mensajes hacia otros usuarios integra aplicativos y sistemas informáticos que permiten presentar y manipular información desde cualquier parte del mundo y a través de cualquier dispositivo que tenga acceso a internet.

GWT (Google Web Toolkit ) es un framework para el desarrollo de aplicaciones Web con interfaz gráfica de usuario enriquecida (RIA) basadas en Ajax (Asynchronous JavaScript and XML, JavaScript asíncrono y XML), por medio del lenguaje de programación Java, dado que Ajax basa su funcionamiento en lenguajes tipo script, GWT compila el código Java de la aplicación a su equivalente JavaScript.

GWT proporciona APIs que facilitan la creación de interfaces gráficas de usuario de forma similar a las librerías AWT o Swing de Java, este ofrece diversidad de componentes gráficos que van desde manejadores de espacio (layouts), hasta listas y tablas para la muestra de datos, además de facilitar las modificaciones de la apariencia de la aplicación sin tener que volver a compilarla.

Una característica de GWT es que, al estar basada en Ajax, su ejecución se realiza en el cliente por medio del navegador Web, lo cual permite que el control de la interfaz gráfica de la aplicación sea manejado allí y no en el servidor como lo hacen las aplicaciones Web tradicionales; esto trae la ventaja del hecho de tener que transportar con la interfaz de usuario a mostrar solo la información necesaria para su funcionamiento, lo cual a su vez libera la carga en el servidor y ofrece una mayor interactividad y usabilidad en este tipo de aplicaciones. [2]

Dentro de los posibles sistemas de identificación de personas que existen se escogió la biometría que es un método automático de reconocimiento de individuos, basado en características biológicas (anatómicas y psicológicas) y de comportamiento que se pueden medir y que puede ser utilizada en el reconocimiento automática.

Las técnicas biométricas implementadas o estudiadas incluyen huellas dactilares, rostro, iris, voz, firma y geometría de la mano, dentro de la biometría muchos factores deben tenerse en cuenta al implementar un dispositivo biométrico, incluyendo la ubicación, los riesgos de seguridad, la tarea (de identificación o de verificación), cantidad de usuarios esperables, circunstancias de utilización, datos existentes, etc.

Es también importante notar que las modalidades biométricas están en distintas etapas de maduración. Por ejemplo, el reconocimiento por huellas dactilares ha sido utilizado por más de un siglo dando excelentes resultados, mientras que el reconocimiento por iris no tiene más de una década de utilización. Debe tenerse en cuenta también que la madurez del dispositivo no está relacionada con cuál de ellos es el mejor, pero puede ser un indicador de las tecnologías que tienen mayor experiencia en la implementación.

Las tecnologías biométricas están siendo utilizadas en múltiples lugares para ampliar la seguridad y comodidad de una sociedad. Además de organismos de Gobierno, muchas empresas privadas están implementando tecnologías biométricas para asegurar distintas áreas, llevar registros de tiempo y fortalecer la conveniencia de los usuarios. [3]

El impacto del sistema será de gran beneficio, y se justifica desde el punto de vista económico y tecnológico, ya que constituye una herramienta informática para la automatización y optimización del control de cumplimiento de las jornadas laborales en los horarios establecidos para incrementar la productividad actual de la fábrica y asegurar el correcto uso y consumo de los recursos que posee, reduciendo los gastos operativos y de mantenimiento.

### **Análisis de factibilidad**

La propuesta presentada es factible de aplicarse ya que cuenta con el recurso humano, financiero, y la predisposición de las personas involucradas, siendo un caso de empleo de la tecnología GWT en la creación de un sistema web eficaz y eficiente adaptándose fácilmente al ambiente laboral.

## **1.5.Objetivos**

### **1.5.1. Objetivo General**

Implementar tecnología GWT en la creación de un sistema web de control de asistencia usando biometría dactilar para la fábrica textil GMB ubicada en el cantón Pelileo.

### **1.5.2. Objetivos Específicos**

- Analizar la actual situación del control de asistencia del personal en la fábrica.
- Analizar los recursos tecnológicos que posee la fábrica para implementar el software desarrollado con GWT.
- Investigar las características, arquitectura, ventajas, desventajas y requerimientos de software y hardware de GWT para ser implementado en un ambiente controlado.
- Diseñar y desarrollar la aplicación web para el control de asistencia del personal.
- Implantar la aplicación desarrollada con tecnología GWT.

## CAPITULO 2

### MARCO TEORICO

#### 2.1. Antecedentes Investigativos

En el presente proyecto se propone explotar las ventajas del framework GWT en el desarrollo de una aplicación web, actualmente esta tecnología no ha sido lo suficientemente explotada en el desarrollo de aplicaciones RIAs para la web 2.0, existen algunos proyectos donde se ha implementado con éxito dicha tecnología logrando excelentes resultados.

“Programando redes de sensores inalámbricos en la forma Web 2.0”. En este proyecto se muestra la implementación de la tecnología GWT en la creación de un sistema para la web 2.0 que administra una red de sensores y analiza los datos que estos generan, proporciona un conjunto de herramientas para el desarrollo de aplicaciones de WSN, en forma similar a lo que GWT ofrece para aplicaciones Web. Se implementa el componente que genere la aplicación WSN (WSA) y la aplicación que se ejecuta en Internet en Java (IAP) de la manera realizada actualmente para servidor y navegador en la Web. [4]

“Método para Adaptar una Librería de Processing a la Web”. En este proyecto se presenta un método para adaptar librerías de Processing a Processing.js, sin conocimientos de JS y manteniendo un solo trunk de desarrollo.

Las librerías adaptadas en este artículo son solo pruebas de concepto que demuestran que desarrollando más el método a futuro se podrá adaptar librerías con mucha complejidad como Proscene. Como trabajo futuro, se podría automatizar este método en una herramienta para el IDE de Processing o para Eclipse IDE; sin embargo las limitaciones del método están dadas por el uso de GWT, diferencias irreconciliables entre los lenguajes Java y JS como, hilos, reflection y demás; pero pueden ser abordados con nuevas tecnologías como HTML5 Web Workers and extensions. [5]

“Análisis de la plataforma RIA GWT para desarrollo en AJAX para el departamento de recursos humanos de la refinería estatal de esmeraldas”. Este proyecto muestra la creación de una aplicación web compilada con el framework GWT, dicha aplicación se encarga de administrar la información más relevante del personal que trabaja en la refinería estatal de Esmeraldas, Provincia de Esmeraldas, Ecuador, Este proyecto es un claro ejemplo de como el framework GWT facilita el proceso de desarrollo de una aplicación web mejorando además el diseño y presentación de las interfaces para hacerlas mas amigables para los usuariosque se benefician de esta aplicacion. [6]

“Web Application Deveploment Technologies Using Google Web Toolkit And Google App”. Este artículo muestra que el uso de patrones de diseño en la construcción de la interfaz gráfica, ayudan en la obtención de un software de calidad, extensible y reutilizable, a la vez de solucionar los problemas para los que fueron diseñados.

Debido a la naturaleza en la forma en que funcionan las tecnologías GWT y App Engine y a las restricciones que tienen, su integración con otros frameworks para Java, como por ejemplo Spring, es algo complejo y no se pueden aprovechar todos las beneficios ofrecidos por estos. [7]

“Integración de APIs políglotas de mapas en Google Web Toolkit: IDELabMapstractionGWT ”, En este artículo muestra que desarrollo



IDELabMapstraction GWT ofrece una nueva posibilidad de crear mapas apoyados en el API Universal de mapas Mapstraction. Ahora estos mapas se pueden integrar dentro de cualquier aplicación que se desee crear utilizando Google Web Toolkit, por lo que el espectro de desarrolladores que tendrán la posibilidad de acceder a las ventajas que ofrece este API se verá aumentado considerablemente.

Además, se ofrecen varios componentes gráficos que se pueden asociar fácilmente a los mapas y que ofrecen la posibilidad de interactuar con el mapa de forma sencilla. Las ventajas que ofrece el hecho de programar en Java sobre GWT en lugar de hacerlo sobre el Javascript nativo son numerosas, pero principalmente caben destacar las ventajas de utilizar un lenguaje tipado y realmente orientado a objetos para llevar a cabo los desarrollos. El widget GWT que representa los mapas es totalmente integrable dentro de cualquiera de los elementos que ofrece la interfaz de usuario de GWT. Además, se han desarrollado una serie de componentes auxiliares para los mapas que facilitan las labores de los desarrolladores que deseen crear componentes auxiliares para los mapas, como pueden ser controles de zoom o para modificar el bounding box. Las ventajas que ofrece el modo de depuración hosted son visibles por la sencillez con la que se depuran estas aplicaciones comparado con las herramientas de depuración de código Javascript. [8]

“Propuesta de integración de Google Web Toolkit con Glassfish y Enterprise Java Bean para desarrollar un sistema web y la administración de información” En el presente artículo se presenta una propuesta de integración de algunas de las tecnologías más sobresalientes en el mundo de la informática, esto como resultado del desarrollo de un sistema web sobre la plataforma de Google Web Toolkit, obteniendo la integración de Google Web Toolkit con Glassfish y Enterprise Java Bean, para que sirva como guía básica para aquellas equipos de desarrollo de software que se inician en el uso de la tecnología Google Web Toolkit.

Además se presenta los resultados del desarrollo de un sistema para administrar información haciendo uso de GWT(Google Web Toolkit) con Glassfish y Enterprise Java Bean, según el plan de pruebas que comparo los resultados obtenidos al implantar

el sistema web (plan basado en pre-test y post-test , administración de información sin y con el sistema web). [9]

## **2.2. Fundamentación Teórica**

### **2.2.1. Lenguaje Java**

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Con respecto a la memoria, su gestión no es un problema ya que ésta es gestionada por el propio lenguaje y no por el programador.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por *hardware* del *bytecode* por un procesador Java también es posible. [10]

### **2.2.2. JDK**

Java Development Kit, es un software que provee herramientas de Desarrollo para la creación de programas en Java. Puede instalarse en una Computadora local o en una unidad de Red. [11]

### **2.2.3. Desarrollo de Software**

“El desarrollo de software no es una tarea solamente técnica, en la cual lo único que importa es la tecnología y los desarrolladores. La producción de software generalmente también involucra a terceros (es decir, en la mayoría de situaciones se desarrolla un programa para satisfacer una necesidad específica de un usuario que no es el mismo programador). Por lo tanto, el éxito de un programa está sujeto a que este haga, que haya sido desarrollado con los recursos estimados y que sea confiable.” [12]

#### **2.2.4. Tecnología WEB**

Las tecnologías Web sirven para acceder a los recursos de conocimiento disponibles en Internet o en las intranets utilizando un navegador. Están muy extendidas por muchas razones: facilitan el desarrollo de sistemas de Gestión del Conocimiento, su flexibilidad en términos de escalabilidad, es decir, a la hora de expandir el sistema; su sencillez de uso y que imitan la forma de relacionarse de las personas, al poner a disposición de todos el conocimiento de los demás, por encima de jerarquías, barreras formales u otras cuestiones. Estas tecnologías pueden llegar a proporcionar recursos estratégicos, pero, evidentemente, no por la tecnología en sí misma, que está disponible ampliamente, sino por lo fácil que es personalizarla y construir con ella sistemas informáticos para la empresa. [13]

#### **2.2.5. Desarrollo Web en Java**

Las páginas JSP son una solución de tercera generación que se pueden combinar fácilmente con algunas soluciones de la segunda generación, creando el contenido dinámico, y haciendo más fácil y más rápido construir las aplicaciones basadas en Web que trabajan con una variedad de otras tecnologías: servidores Web, navegadores Web, servidores de aplicación y otras herramientas de desarrollo. La tecnología Java Server Pages™ (JSP) permite poner segmentos de código servlet directamente dentro de una página HTML estática.

Cuando el navegador carga una página JSP, se ejecuta el código del servlet y el servidor de aplicaciones crea, compila, carga y ejecuta un servlet en segundo plano para ejecutar los segmentos de código servlet y devolver una página HTML o imprimir un informe XML. [14]

#### **2.2.6. Arquitectura WEB**

Las aplicaciones web utilizan lo que se conoce como clientes livianos (light clients) los cuales no ejecutan demasiadas labores de procesamiento para la ejecución de la aplicación misma. Desde el punto de vista de la arquitectura se distinguen dos lados; uno es el cliente, donde se encuentra el usuario final utilizando la aplicación por medio de un

navegador (como Internet Explorer o Mozilla Firefox). A través de este cliente web, el usuario interactúa con la aplicación localizada al otro lado, en el servidor, que es donde residen realmente los datos, reglas y lógica de la aplicación. [15]

### **Arquitectura de tres niveles**

También conocida como arquitectura de tres capas, la arquitectura de tres capas, define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además ayuda a identificar qué puede reutilizarse, y proporciona una estructura que ayuda a tomar decisiones sobre qué partes comprar y qué partes construir.

Para enfrentarse a estos temas, la comunidad de software desarrolló la noción de una arquitectura de tres niveles. La aplicación se divide en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definido. La primera capa se denomina capa de presentación y normalmente consiste en una interfaz gráfica de usuario de algún tipo.

La capa intermedia, o capa de empresa, consiste en la aplicación o lógica de empresa, y la tercera capa, la capa de datos, contiene los datos necesarios para la aplicación. La capa intermedia (lógica de aplicación) es básicamente el código al que recurre la capa de presentación para recuperar los datos deseados. La capa de presentación recibe entonces los datos y los formatea para su presentación.

Esta separación entre la lógica de aplicación de la interfaz de usuario añade una enorme flexibilidad al diseño de la aplicación. Pueden construirse y desplegarse múltiples interfaces de usuario sin cambiar en absoluto la lógica de aplicación siempre que está presente una interfaz claramente definida a la capa de presentación.

### **Capa de presentación.**

Es la que se encarga de que el sistema interactúe con el usuario y viceversa, muestra el sistema al usuario, le presenta la información y obtiene la información del usuario en un mínimo de proceso. En el mundo de la informática es conocida como interfaz gráfica y debe tener la característica de ser amigable, o sea, entendible y fácil de usar para el usuario. Esta capa se comunica únicamente con la capa intermedia o de negocio.

### **Capa de negocio**

Es donde residen las funciones que se ejecutan, se reciben las peticiones del usuario, se procesa la información y se envían las respuestas tras el proceso. Se denomina capa de negocio o capa de lógica del negocio, porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

### **Capa de acceso a datos**

Esta capa es la encargada de almacenar los datos del sistema y de los usuarios. Su función es almacenar y devolver datos a la capa de negocio, aunque para esto también es necesario en algunos casos, que tengan procedimientos almacenados y funciones dentro de la capa. En una arquitectura de tres capas, esta capa es la única que puede acceder a los mismos. Está formada por uno o varios sistemas gestores de bases de datos, localizados en un mismo servidor o en varios.

Estas capas, pueden estar localizadas todas en un mismo ordenador, si el programa o software informático que se desarrolla es de baja complejidad, porque si, por el contrario, fuera de gran complejidad tanto los datos como la lógica de negocio, entonces cada una de las capas pudiera estar situada en diferentes ordenadores, para mejorar la funcionalidad de las mismas, incluso, en productos de gran complejidad, existen varios ordenadores para la capa de acceso a datos, y varios ordenadores para la capa de negocio. [16]

### **2.2.7. IDE de desarrollo**

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. [17]

### **2.2.8. Ajax**

Realizar peticiones al servidor y esperar respuesta puede consumir tiempo (el tiempo necesario para recargar una página completa). Para agilizar los desarrollos web surgió una tecnología que busca evitar las demoras propias de las peticiones y respuestas del servidor mediante la transmisión de datos en segundo plano usando un protocolo específicamente diseñado para la transmisión rápida de pequeños paquetes de datos.

Con Ajax, se hace posible realizar peticiones al servidor y obtener respuesta de este en segundo plano (sin necesidad de recargar la página web completa) y usar esos datos para, a través de JavaScript, modificar los contenidos de la página creando efectos dinámicos y rápidos. [18]

### **2.2.9. Bases de Datos**

Una base de datos (cuya abreviatura es BD) es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Diferentes programas y diferentes usuarios deben poder utilizar estos datos. Por lo tanto, el concepto de base de datos generalmente está relacionado con el de red ya que se debe poder compartir esta información. De allí el término base. "Sistema de información" es el término general utilizado para la estructura global que incluye todos los mecanismos para compartir datos que se han instalado.

Una base de datos proporciona a los usuarios el acceso a datos, que pueden visualizar, ingresar o actualizar, en concordancia con los derechos de acceso que se les hayan

otorgado. Se convierte más útil a medida que la cantidad de datos almacenados crece. [19]

“Las aplicaciones web reciben este nombre porque se ejecutan en la internet. Es decir que los datos o los archivos en los que se trabaja son procesados y almacenados dentro de la web. Estas aplicaciones, por lo general, no necesitan ser instaladas en un computador.

El concepto de aplicaciones web está relacionado con el almacenamiento en la nube. Toda la información se guarda de forma permanente en grandes servidores de internet y se envían a nuestros dispositivos o equipos los datos que se requieren en ese momento, quedando una copia temporal dentro del equipo.

Estos grandes servidores de internet que prestan el servicio de alojamiento están ubicados alrededor de todo el mundo, así hacen que el servicio prestado no sea tan costoso o gratuito en la mayoría de los casos y extremadamente seguro.” [20]

#### **2.2.10. RIAs**

Aplicaciones Ricas de Internet (Rich Internet Applications, RIAs) o Aplicaciones Web. Una aplicación web va mucho más allá del enfoque de una página web estándar, llega al nivel de las aplicaciones de escritorio e incluso a veces las superan, en términos de arquitectura de software, las aplicaciones web suelen dividirse en:

- La capa de **Vista o Interfaz**, que es la que el usuario ve en su navegador y mediante la cual interactúa: es donde se encuentran las ventanas, botones, menús, imágenes, etc.
- La capa de **Lógica de Negocio** (o Servicios de Negocio), es la encargada de ejecutar la acción que el usuario solicita (pulsando un botón, seleccionando una opción del menú) y de informar del resultado de la misma.

La capa de interfaz se maneja desde la máquina local del usuario, mediante su navegador web, mientras que la lógica de negocio está centralizada en un servidor.

Una aplicación de escritorio se ve como un paquete que, mediante instalación o sin ella, se ejecuta de forma íntegra en el ordenador de cada uno. [21]

### **2.2.11. Google Web Toolkit**

Google Web Toolkit (GWT) es un framework de código abierto desarrollado por Google y lanzado en mayo del 2006 bajo la licencia Apache 2.0. con el objetivo de facilitar el desarrollo de RIAs, que permite escapar de la “matriz” de tecnologías usadas actualmente para escribir aplicaciones AJAX, las cuales son difíciles de manejar y propensas a errores. El corazón de GWT es un compilador que compila código JavaScript a partir de código Java. El compilador es capaz de generar código multibrowser, aliviando la tarea del desarrollador de escribirlo por sí mismo. GWT genera código interpretable por los browser más importantes del mercado (Mozilla Firefox, Safari, Opera e Internet Explorer).

### **2.2.12. Características de GWT:**

El compilador GWT es justamente un mecanismo para entregar enteramente una aplicación arquitectura cliente/servidor.

Las características incluyen:

- Componentes de la interfaz de usuario dinámico y reutilizables (widgets). Y RPC realmente fácil.
- Depuración en tiempo real (mientras lo estás desarrollando este corre sobre una máquina virtual de Java).
- Compatibilidad con los navegadores
- Integración con JUnit
- Internacionalización
- Interoperabilidad y control. (Si las librerías de clases de GWT no son suficientes para lo que se necesita, se puede mezclar JavaScript en el código de la aplicación usando la interfaz nativa de scripts de Java (JavaScript Native Interface, JSNI)).
- GWT es un proyecto de código abierto. (Disponible bajo la licencia Apache 2.0)
- Soporte para las API's de Google
- Se puede usar los entornos de desarrollos integrados (IDE) de siempre.
- Comunicación entre el cliente y servidor usando objetos de java.



- Los desarrolladores pueden diseñar y desarrollar sus aplicaciones orientadas a objetos.
- El código Javascript generado puede ser ofuscado para optimizar el rendimiento.
- Existe un numeroso conjunto de bibliotecas desarrolladas por Google y terceros que amplían las funcionalidades de GWT.

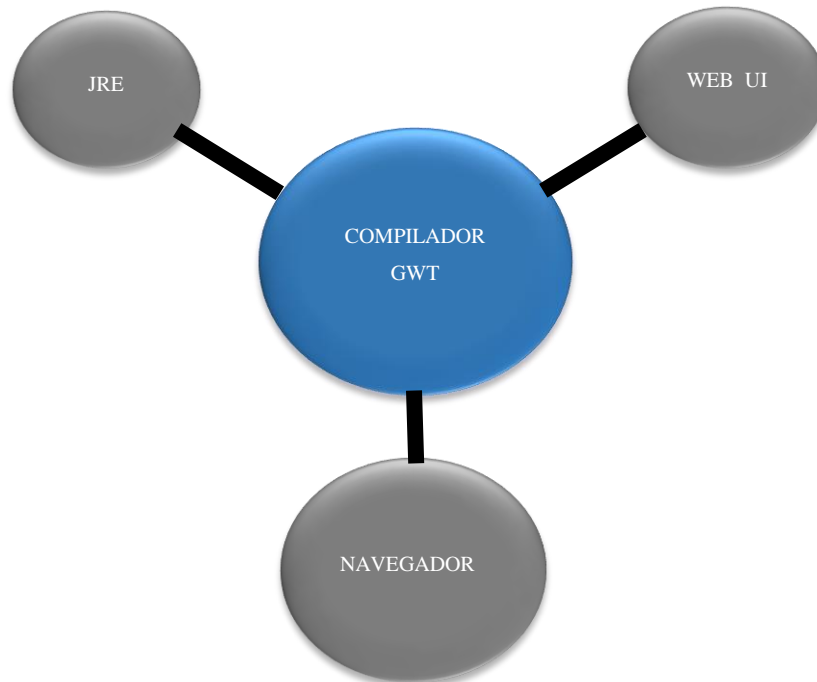
### 2.2.13. Arquitectura GWT

GWT tiene cuatro componentes principales: un compilador Java-a-JavaScript, un navegador web "hosted", y dos librerías de clases.

Componentes:

- GWT Java-to-JavaScript Compiler (Compilador GWT Java-a-JavaScript): El Compilador GWT Java-a-JavaScript traduce del lenguaje de programación Java a JavaScript.
- Hosted Web Browser (Navegador web "Hosted" de GWT): El Navegador web "Hosted" de GWT permite correr y ejecutar aplicaciones GWT en modo hosted, donde lo que se está corriendo son bytecodes de Java sobre una máquina virtual sin compilarlos a JavaScript. Para lograr esto, el navegador GWT incrusta un controlador de browser especial (un control del Internet Explorer sobre Windows o un control de Gecko/Mozilla sobre Linux) con hooks dentro de la máquina virtual de Java.
- JRE Emulation Library (Emulación de librerías JRE): Google Web Toolkit soporta solamente una pequeña parte de las librerías de clases disponibles en Java 2 Standard Edition y Enterprise Edition, esto sucede ya que éstas librerías son bastante grandes y usan funcionalidades que no están disponibles en navegadores web.
- GWT Web UI Class Library (Librería de clases de interfaz de usuario de GWT): Las librerías de clases de interfaz de usuario de GWT son un conjunto de interfaces y clases personalizadas que permiten crear "widgets" para el navegador, como botones, cajas de texto, imágenes, etc. Éste es el

núcleo de las librerías de interfaz de usuario para crear aplicaciones GWT.  
[9]



*Imagen 1. Arquitectura GWT. Elaborado por: El Investigador*

#### **2.2.14. Biometria**

El termino biometría viene del griego "bio" que significa vida y "metría" que significa medida o medición, de acuerdo al diccionario de la real academia de la lengua española biometría es el estudio mensurativo o estadístico de los fenómenos o procesos biológicos, sin embargo más recientemente y para el tema que concierne el significado de biometría es el conjunto de métodos automatizados que analizan determinadas características humanas para identificar o autenticar personas. [3]

#### **2.2.15. Biometría Dactilar**

Una huella dactilar usualmente aparece como una serie de líneas oscuras que representan los relieves, la porción saliente de las crestas de fricción, mientras los valles entre estas

crestas aparecen como espacio en blanco y están en bajo relieve, la porción subyacente de las crestas de fricción.

La identificación por huella dactilar está basada principalmente en las minucias, o la ubicación y dirección de los finales y bifurcaciones (separaciones) de las crestas a lo largo su trayectoria.

Las imágenes debajo presentan ejemplos de características de huellas dactilares:

(A) Dos tipos de minucias y (B) ejemplos de otras características algunas veces utilizadas durante la clasificación automática y procesos de extracción de minucias:

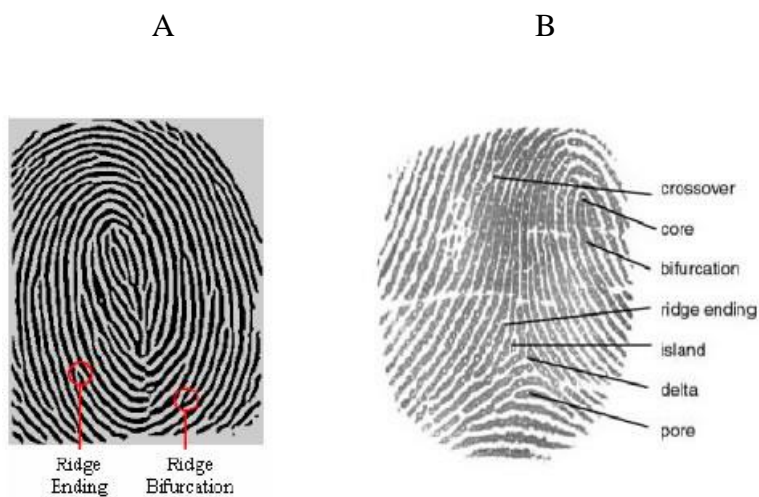


Imagen 2. Huellas Dactilares. Elaborado por: <http://www.biometria.gov.ar/metodos-biometricos/dactilar.aspx>

[3]

### **2.2.16. Propuesta de Solución**

La propuesta para solucionar el problema planteado es Aplicar la tecnología GWT en el desarrollo de una aplicación que permitirá automatizar el control de asistencia del personal de la fábrica textil GMB, incrementando la productividad de la fábrica reduciendo el consumo excesivo de recursos.

Este sistema contará con varios módulos que cubrirán cada uno de los problemas existentes, se mejorara el control de asistencia y cumplimiento de las jornadas laborales en los horarios de trabajo establecidos optimizando la generación de roles de pago.

## **CAPITULO 3**

### **METODOLOGÍA**

#### **3.1. Modalidad de la Investigación**

En esta investigación se destina la investigación proyectiva, ya que se ha elaborado una propuesta o modelo para solucionar un problema.

Modalidad Bibliográfica o Documentada: Se ha optado utilizar esta modalidad por que la información se ha obtenido de Internet, Libros virtuales, Tesis, Artículos publicados en la Web, Libros, etc.

Modalidad Campo y Documental: Se ha optado utilizar esta modalidad por que el investigador se apoyará en documentos y registros físicos y virtuales para implementar la tecnología que es caso de estudio para posteriormente recoger la información directamente de las personas involucradas con la futura solución informática.

#### **3.2. Recolección de Información**

Para comprender y posteriormente aplicar la tecnología GWT se obtendrá información mediante consultas en repositorios virtuales oficiales donde se encuentren registrados la documentación que explique la forma correcta de aplicar dicha tecnología.

Se recogerá información de diferentes maneras, por ejemplo, observando los problemas que existen actualmente, mediante documentos, archivos y si es necesario internet para conocer problemas parecidos a lo que se tiene, se realizará entrevistas o cuestionarios a las personas a cargo de estos procesos y los que forman parte de los procesos.

Esta recolección de información se realizará para conocer los problemas que se tiene y así poder solucionarlos de acuerdo al cronograma de actividades y se realizara las veces que sea necesario en el lugar de los hechos, hasta tener clara y completa la información necesaria para el desarrollo del proyecto.

### **3.3. Procesamiento y Análisis de Datos**

De acuerdo a la información obtenida se aplicará el siguiente procedimiento para el procesamiento y análisis de los datos.

- Obtener los recursos necesarios para implementar la tecnología GWT.
- Elección de la metodología correcta para desarrollar una aplicación web con GWT.
- Crear el ambiente tecnológico adecuado para que se pueda utilizar la tecnología GWT correctamente.
- Analizar la información necesaria para automatizar los procesos de control de asistencia.
- Realizar una entrevista a las personas encargadas del proceso de control de asistencia. De esta manera se optimizara dicho proceso.

### **3.4. Desarrollo del Proyecto**

Para poder cumplir con los objetivos del desarrollo de este proyecto se utilizará una metodología muy conocida en el desarrollo de Software llamada “Cascada” en donde cumplen las diferentes fases para el desarrollo que son: Análisis, Diseño, Codificación, Prueba, Verificación y Mantenimiento. De acuerdo a esto las actividades a desarrollar en forma secuencial serán las siguientes:

- Obtener información relevante al uso y funcionamiento de la tecnología GWT
- Preparación del ambiente de desarrollo para hacer uso de GWT
- Identificación de los procesos que existen en el control de asistencia y cumplimiento de las jornadas laborales.

- Análisis del tipo de información que se maneja en cada uno de los procesos identificados.
- Análisis y selección de las tecnologías y herramientas a utilizar en el desarrollo de la aplicación web.
- Análisis y selección de personal que respalde de la mejor manera al proyecto.
- Diseño de organigramas, flujo gramas, procesos nuevos o reestructurados, base de datos, interfaz gráfica de la aplicación, plantillas, etc.
- Desarrollo de la aplicación web siguiendo normas, estándares, metodologías, etc., para el buen desarrollo de la aplicación web.
- Pruebas constantes durante el desarrollo de la aplicación para ir corrigiendo errores.
- Diseño de un aplicativo o manual de uso de la aplicación web para el buen uso de la aplicación por parte de los usuarios finales.
- Implementación de la aplicación web.

## **CAPITULO 4**

### **DESARROLLO DE LA PROPUESTA**

En el desarrollo del presente proyecto se utilizó el modelo en cascada, mediante el cual se ordenan rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior, de esta manera, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado.

#### **4.1. Análisis del Sistema y Requerimientos**

##### **4.1.1. Recolección y Análisis de Información**

La información necesaria para la realización de este proyecto se recolectó mediante entrevistas, de esta manera se da a conocer la técnica y forma de realizar los procesos de registro de asistencia.

A las personas que se les realizó la entrevista son las involucradas en los procesos como los colaboradores que trabajan en la fábrica textil, la secretaria, el gerente propietario. Cada una de estas personas cumple diferentes funciones y manejan diferentes documentos de registro de información.

Se realizó una entrevista a la secretaria encargada del proceso y control de registros de asistencia para poder conocer de mejor manera el proceso que se pretende automatizar. A continuación, se da un resumen general de la entrevista.



## Análisis de la entrevista

Pregunta	Respuesta
<p><b>¿Qué personas intervienen en el proceso de Registro de la Asistencia?</b></p>	<p>El personal colaborador de planta y personal administrativo.</p>
<p><b>¿Cómo se realiza el proceso de registro?</b></p>	<p>El personal de planta y administrativo registra en un documento impreso sus datos personales, hora de entrada y salida a las diferentes jornadas laborales, certifican su asistencia con una firma, este documento se encuentra en la oficina de recepción, cuando existe concurrencia de colaboradores se organizan en orden de llegada para poder registrarse.</p>
<p><b>¿Cómo están establecidos los horarios laborales?</b></p>	<p>Existen colaboradores que tienen su propio horario debido que sus tareas no se complementan con las tareas de otro colaborador, las demás personas trabajan en horarios normales.</p>
<p><b>¿Qué información se obtiene al momento de registrar la asistencia?</b></p>	<p>Datos personales, hora y fecha del momento exacto del registro, mismo que puede ser entrada o salida de las jornadas laborales.</p>
<p><b>¿Existen excepciones en el proceso de control de asistencia?</b></p>	<p>En ocasiones el personal colaborador solicita permisos especiales o justificaciones para no asistir o para no completar la jornada laboral diaria, dichas horas no laboradas deben ser recuperadas los fines de semana, en el caso de una irregularidad en la asistencia no justificada se procede a realizar los descuentos de regla al final de mes en el rol de pago.</p>
<p><b>¿Qué inconvenientes a tenido al llevar el proceso de registros de asistencia manualmente?</b></p>	<p>Los colaboradores no son cuidadosos con los formatos impresos y los deterioran regularmente, no siempre se registra la hora real de entrada y salida de las jornadas laborales, los formatos impresos se guardan en archiveros hasta el fin de mes para hacer un informe mensual, en ocasiones esos registros se extravían o son sustraídos para evitar las sanciones reglamentarios dentro de la empresa, otro inconveniente es el hecho de que existen familiares directos laborando en la empresa</p>

	y en ocasiones registran a sus familiares o amigos que no asisten o están atrasados para evitar las sanciones que esto conlleva, a veces los colaboradores que están atrasados a su jornada laboral se registran rápidamente en los formatos impresos y sus datos personales y/o firma no son lo suficientemente legibles.
--	--

*Tabla 1. Matriz de entrevista realizada a la persona encargada del proceso a ser automatizado. Elaborado por el colaborador.*

## **Conclusión de la entrevista**

Basado en los resultados obtenidos en la entrevista se puede concluir que:

- Las personas que intervienen en el proceso son los colaboradores de planta y administrativos, siendo la secretaria la encargada de la correcta ejecución del proceso de control de asistencia
- El proceso manual es tedioso, repetitivo y redundante en el registro de información personal.
- Los formatos impresos representan un desperdicio de recursos de oficina.
- El formato de registro de entradas y salidas de las jornadas brindan la posibilidad de suplantación de identidad.
- Los registros de tiempo de llegada y salida de las jornadas laborales no pueden brindar la veracidad necesaria para ser completamente válidos.
- El tiempo para registrar la asistencia de los colaboradores es innecesariamente alto debido a la cantidad de colaboradores que deben esperar su turno para poder registrarse en los formatos impresos.
- El proceso manual no asegura la veracidad y disponibilidad de la información necesaria para realizar los informes de asistencia mensuales.

## **Proceso de registro de asistencia**

Actualmente se lleva el registro de asistencia en hojas de papel impresas donde se registra manualmente los datos personales del colaborados que procede a registrarse, la hora de ingreso o salida de las jornadas laborales y para certificar la asistencia el personal colaborador firma en dicha hoja impresa, en el Anexo N°1 se puede observar la

hoja impresa del registro, una vez que el colaborador termina de registrarse en los formatos impresos ingresa a su lugar de trabajo para cumplir con las tareas diarias, la secretaria encargada del control de este proceso, valida la información, en caso de existir atrasos, faltas u otra anomalía injustificada procede a registrar la sanción definida por la gerencia, una vez validado la información se archiva el registro diario, el fin de cada mes se procede a realizar un informe mensual con la información resumida de los registros diario de asistencia.

A continuación, se detalla de manera gráfica en la imagen 3 el flujo de las actividades para realizar el proceso descrito en el párrafo anterior.

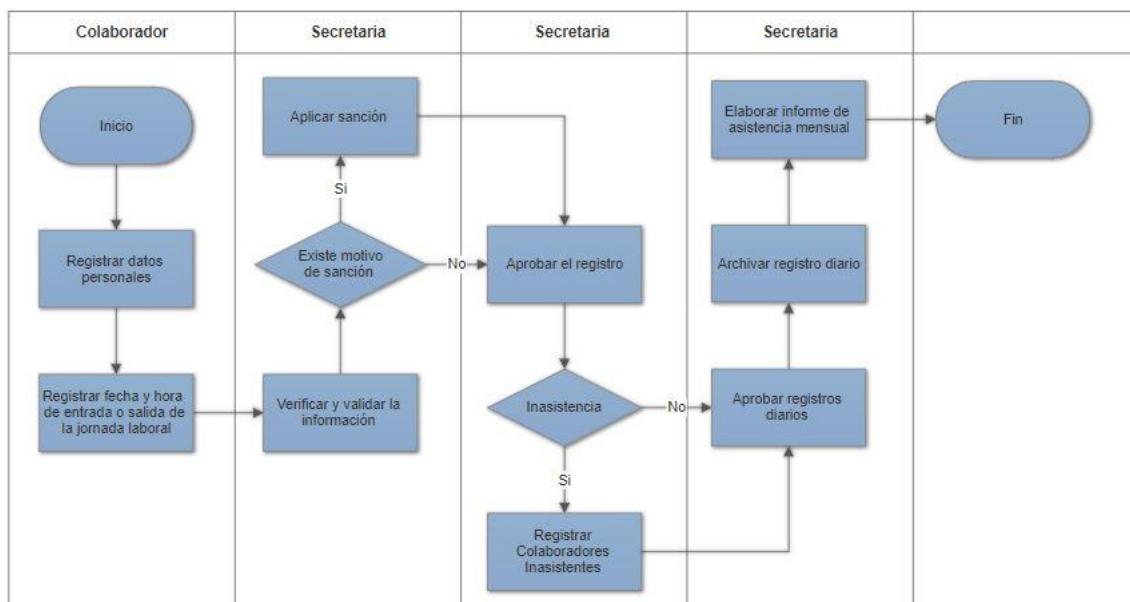


Imagen 3. Diagrama de flujo de procesos del control de asistencia. Elaborado por: El Investigador

#### 4.1.2. Levantamiento de los Requerimientos del Sistema

Con la información recolectada, en la entrevista se establecieron los requerimientos y funcionalidades que debe cumplir el sistema para satisfacer las necesidades encontradas, los requerimientos y procesos más importantes que contará el sistema son:

### Requerimientos de la aplicación web

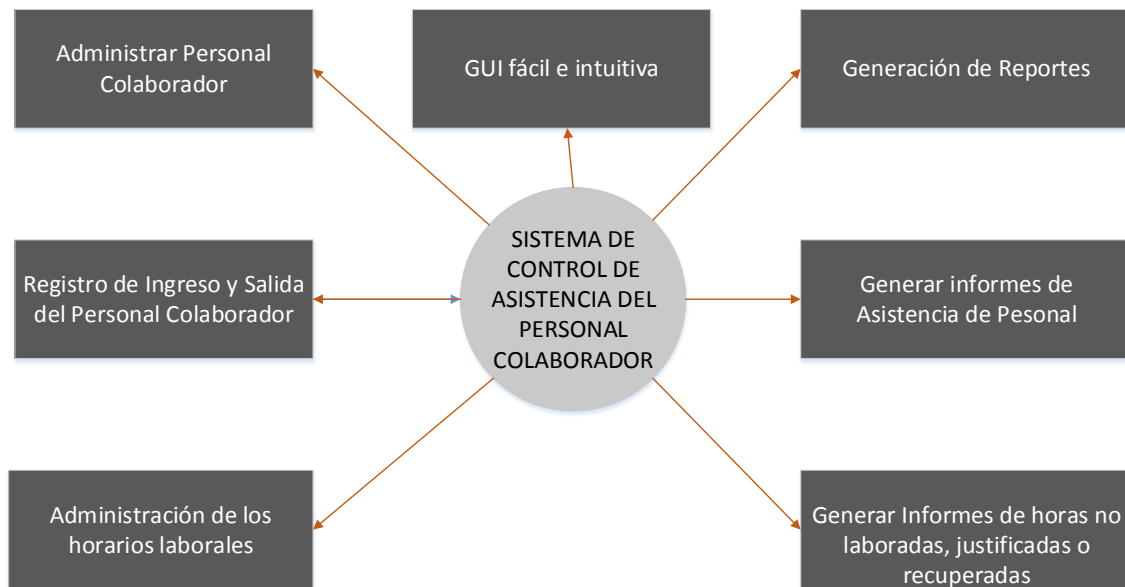


Imagen 4. Requerimientos de la aplicación web. Elaborado por: El Investigador

#### 4.1.3. Selección de Herramientas para el Desarrollo de la Aplicación Web

En este caso se pretende investigar e implementar los beneficios de la tecnología GWT aplicados al desarrollo de una aplicación web.

#### Arquitectura de desarrollo de GWT

El uso de tecnología GWT para desarrollar una aplicación web supone una organización de las funcionalidades de las mismas separando la parte de la aplicación que se va a ejecutar en el cliente y la parte de la aplicación que se ejecutará en el servidor, tomando esto en cuenta, es conveniente estructurar la aplicación GWT de tal manera que todo el manejo de la interfaz gráfica (creación y control de componentes gráficos y manejo de eventos), sea realizada en el cliente, mientras que la manipulación y almacenamiento de la información sea una tarea realizada en el servidor.

Esta estructura mejora la operación de la aplicación ya que optimiza los tiempos de respuesta y mejora la interacción del usuario con la aplicación.

Para crear interfaces graficas se puede hacer uso de diversos patrones de diseño que solucionan problemas como la interacción entre vistas y la independencia entre la forma en que se muestran los datos de las acciones generadas como respuesta a un evento.

Entre estos patrones de diseño se destacan Model View Controller (MVC) y Model View Presenter (MVP) [22], durante el transcurso de los años estos patrones han arrojado buenos resultados en el control del comportamiento de la interfaz gráfica.

En el lado del servidor, se manipula y almacena la información que necesita la aplicación para funcionar, por medio de App Engine, es posible hacer uso de la infraestructura de Google para realizar dichas tareas. Dado que App Engine puede usarse únicamente por los lenguajes de programación Java y Python (aunque también soporta lenguajes interpretados como JavaScript y Ruby), en este trabajo se hace referencia a su versión para Java.

En contexto a lo explicado, la arquitectura sugerida para el desarrollo de aplicaciones GWT y AppEngine-Java se muestra en la imagen 5.

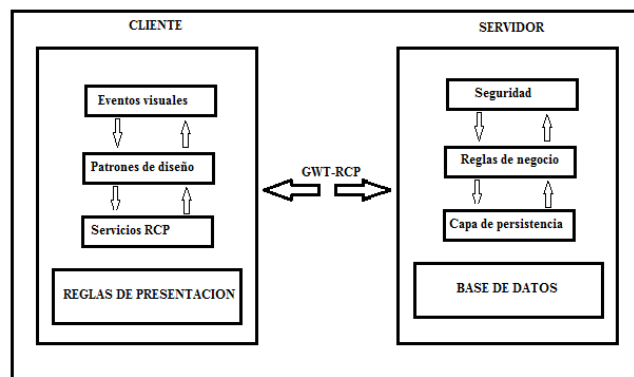


Imagen 5. Figura de la arquitectura sugerida para desarrollar aplicaciones con GWT. Elaborado por el investigador.

La arquitectura sugerida para aplicaciones GWT, en el cliente se controla todo lo relacionado con la capa de presentación y las definiciones de los servicios para la comunicación con el servidor. Por el lado del servidor se encuentran las reglas de negocio, el control de acceso a datos, la seguridad y la implementación de los servicios para el intercambio de información con el cliente, se sugiere emplear GWT-RPC para la

comunicación entre el cliente y el servidor, este permite la transferencia directa de objetos, facilitando su manipulación.

### Ventajas y desventajas del Framework GWT

Ventajas	Desventajas
Facilita el diseño y creación de interfaces de usuario graficas brindando la posibilidad de aplicar características como drag & drop.	Genera demasiado código HTML para implementar y mostrar componentes gráficos incrementado el tiempo de respuesta para la visualización de objetos.
La integración de JSON y XML permite la implementación y la interacción de varias tecnologías en el lado del servidor.	La creación de interfaces graficas simples implica una sobreutilización de recursos.
GWT-RCP como método de comunicación entre el cliente y el servidor facilita el proceso de transmisión de información entre los mismos.	Los llamados al servidor por ser asíncronos pueden ocasionar que los datos no se presenten de manera correcta para su manipulación.
Existen varias herramientas y librerías graficas que ayuda a personalizar y optimizar el diseño de las interfaces graficas	Al compilar la aplicación web con GWT y obtener JavaScript la hace propensa a ataques de inyección de Scripts (XSS) y complica la indexación y rastreo de la aplicación a través de motores de búsqueda
Una aplicación compilada con GWT retorna archivos JavaScript permitiéndola ser ejecutada en cualquier navegador que soporte JavaScript sin necesidad de usar plugins u otro programa adicional	

Tabla 2. Cuadro comparativo de ventajas y desventajas del framework GWT. Elaborado por el investigador.

## Ventajas y desventajas del Framework AppEngine-Java

Ventajas	Desventajas
Brinda la posibilidad de usar la infraestructura de Google en las aplicaciones	La relación del modelo de dominio está sujeto a restricciones.
Ofrece excelente soporte para la ejecución de aplicaciones.	No brinda acceso a todas las funcionalidades definidas por JPA y JDO.
Permite crear modelos de datos a partir de estándares para el acceso a datos persistentes en bases de datos desde aplicaciones Java, por medio del uso de objetos planos Java (POJOs) por medio de las especificaciones JPA (Java Persistent API) y JDO (Java Data Objects).	Restringe el uso de ciertas clases incluida en la máquina virtual de Java JVM
Brinda diversos métodos para acceder a servicios como Memcache, mail, XMPP, autenticación mediante cuentas de Google, entre otras.	Para realizar búsquedas avanzada de texto directamente es necesario realizar algunas adaptaciones al modelo aplicando técnicas para FTS (Full Text Search) .
Facilita la subida de la aplicación a la infraestructura de Google.	

*Tabla 3. Tabla comparativa de ventajas y desventajas del framework AppEngine-Java*

Dicha tecnología utiliza Java como lenguaje de programación, teniendo esto en cuenta a continuación se detallan las herramientas que se usarán en este proyecto de investigación:

- IDE Eclipse Mars 2.
- Google Web Toolkit (GWT) como framework para aplicaciones web.
- Java como lenguaje de programación.

- Componentes Gráficos propios de GWT para tener una interfaz de usuario amigable e intuitiva.
- PostGreSql como base de datos.
- Editor de texto para la edición de códigos en cualquier lenguaje de programación.
- StarUML para diagramas de casos de uso, secuencia, etc.

#### **4.1.4. Definición de los Requerimientos**

Ya realizado un análisis y con un conocimiento claro de los procesos de registro y control de asistencia de los colaboradores a las jornadas laborales se procede a establecer los requisitos técnicos tanto de software como de hardware.

#### **Requerimientos de Hardware**

El requerimiento de hardware se dividirá en dos partes: la del servidor y la del cliente, ya que estos dos ambientes que se manejan de manera diferente, pero para que se ejecute la aplicación correctamente se necesitara que cumpla obligatoriamente con los requerimientos de los dos ambientes:

Servidor:

- 64 MB de memoria RAM.
- 2 GB de espacio en el Disco duro.
- Tarjeta de Red

Dispositivo Biométrico:

- Dispositivo Electrónico que detecte y reconozca huellas Dactilares.
- Cables y conectores para conectar el dispositivo al servidor.

#### **Requerimientos de Software**

Para los requerimientos de software de igual manera se tomarán en cuenta los dos ambientes: el software del servidor y software del cliente. Para cada uno de estos



ambientes se necesita software específico ya que no se puede permitir tener instalado el mismo software en estos ambientes por motivos de seguridad y licenciamiento.

Servidor:

- Sistema Operativo (S.O.).
- Sistema Gestor de Bases de Datos (S.G.D.B.)
- Software contenedor web.
- Controladores para conexión a la base de datos.
- Controlador para conexión con el dispositivo electrónico.
- Máquina virtual Java.

Cliente:

- El único requerimiento para el cliente es tener un navegador instalado que sea compatible con HTML y HTML5 ya que GWT usa ciertos componentes gráficos que se basan en etiquetas HTML5. A continuación en las Imágenes 6 y 7 se describe dos listas con información de compatibilidad de navegadores y sus versiones.

HTML5 Forms Inputs													
	MAC				WIN								
	SAFARI	FIREFOX	OPERA	CHROME	SAFARI	IE				FIREFOX	OPERA	CHROME	
	5.1	11	11.62	18	5.1	6	7	8	9	11	11.61	18	
Form: Search	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	63%
Form: Phone	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	63%
Form: URL	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	61%
Form: Email	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	61%
Form: DateTime	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	4%
Form: Date	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	4%
Form: Month	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	4%
Form: Week	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	4%
Form: Time	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	4%
Form: LocalTime	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	4%
Form: Number	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	30%
Form: Range	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	42%
Form: Colour	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	2%

Imagen 6. Controles de Formulario HTML 5. Elaborado por: <http://desarrolloweb.dlsi.ua.es/>

HTML5 Forms Attributes													
	MAC				WIN								
	SAFARI	FIREFOX	OPERA	CHROME	SAFARI	IE				FIREFOX	OPERA	CHROME	
	5.1	11	11.62	18	5.1	6	7	8	9	11	11.61	18	
Form: Autocomplete	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	31%
Form: Autofocus	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	64%
Form: List	✓	✓	✓	✗	✓	✗	✗	✗	✗	✓	✓	✗	29%
Form: Placeholder	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	63%
Form: Min	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	45%
Form: Max	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	45%
Form: Multiple	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	82%
Form: Pattern	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	63%
Form: Required	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	63%
Form: Step	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	45%

Imagen 7. Atributos de Formulario HTML5. Elaborado por: <http://desarrolloweb.dlsi.ua.es/>

### Recursos tecnológicos que posee la empresa

A continuación, se detalla los recursos tecnológicos que posee la entidad donde será implantada la aplicación web.

Para detallar los recursos tecnológicos que posee la empresa se usara una entrevista como técnica de recolección de información, dicha entrevista fue realizada a la secretaria, misma que se encarga del proceso de control de asistencia.

Pregunta	Respuesta
¿Cuántas PC's posee la empresa?	La empresa dispone de 2 PC's de escritorio y dos portátiles.
¿Cuántas PC's se encuentran operativas y se utilizan diaria o regularmente?	Existen dos PC's de escritorio que se utilizan actualmente. Una PC principal que se encuentra en secretaría y otra que se encuentra en la planta de producción.
¿Qué PC considera usted que posee	La PC de escritorio que se encuentra en la

<b>mejores características en relación a las PC que posee la empresa?</b>	oficina de secretaría. Esa PC tiene las siguientes características Pc core quad i5 4 GB de memoria Ram Disco duro de 1TB
<b>¿Qué PC considera usted que posee las características más básicas en relación a las demás PC?</b>	Una de las portátiles que se encuentra en una sección de la planta de producción se usa muy poco, y se utiliza para controlar parlantes y música dentro de la planta. Es una portátil Toshiba Procesador i3 2 GB de Ram Disco duro de 256 GB

Tabla 4. Entrevista para registrar los recursos tecnológicos existentes en la empresa.

### Resultados de la entrevista

Basado en las respuestas otorgadas en la entrevista se detalla a continuación las PC's que se encuentran en condiciones para alojar la aplicación web.

### Servidor

Hardware	Software
Pc core quad i5 4 GB de memoria Ram Disco duro de 1TB Tarjeta de red	Sistema operativo (S.O.) Máquina virtual Java

## Ciente

<b>Hardware</b>	<b>Software</b>
Portatil Toshiba core quad i3  2 GB de memoria Ram  Disco duro de 256 GB  Tarjeta de red	Sistema operativo (S.O.)  Máquina virtual Java

*Tabla 5. Tablas de recursos tecnológicos que posee la empresa. Elaborado por el investigador*

Dado que la empresa no posee los requerimientos mínimos de software para implementar la aplicación, se procederá a instalar el software necesario para poder implementar la aplicación web.

## 4.2. Diseño de Esquemas de la Aplicación Web

### 4.2.1. Diagramas Casos de Uso

El diagrama en el proceso de creación de software sirve para representar los procesos y el flujo de información que actualmente se maneja dentro de la empresa y así comprender de mejor manera la realidad que se pretende automatizar, la misma que se detalla a continuación en las imágenes 8, 9 y 10:

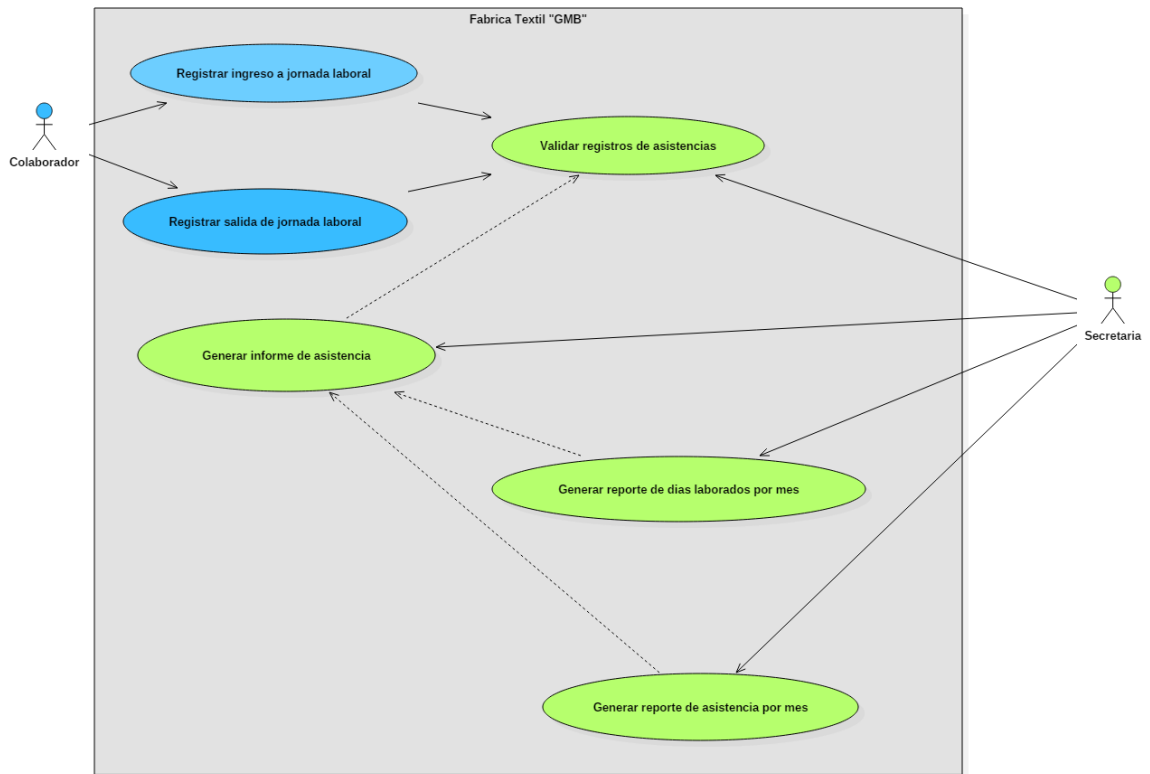


Imagen 8. Casos de Uso. Elaborado: Por el Investigador

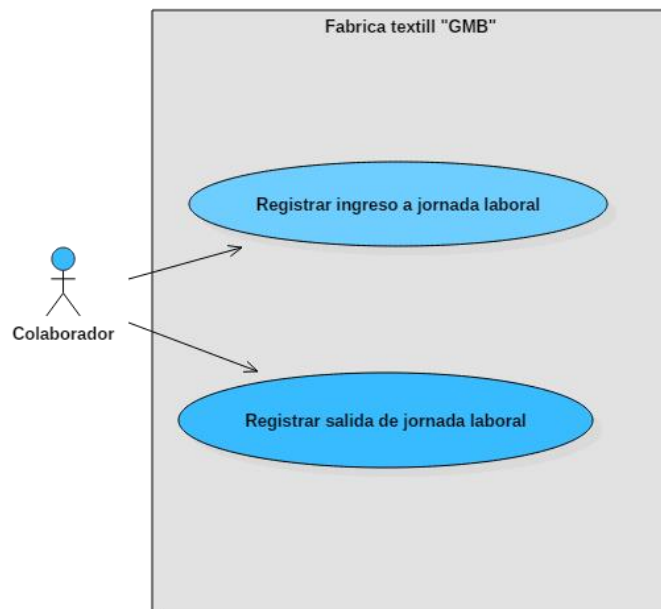


Imagen 9. Caso de Uso Colaborador. Elaborado por el Investigador



Imagen 10. Caso de Uso Secretaria. Elaborado por: El Investigador

#### 4.2.1.1. Descripción de los Actores

Se ha determinado dos actores que cumplirán tareas diferentes y que se complementan en el sistema, estos actores serán los que utilicen el sistema:

<b>Actor 01:</b>	Colaborador
<b>Descripción:</b>	Utiliza el sistema para registrar su asistencia a su lugar de trabajo.

Tabla 6. Descripción del Actor Colaborador. Elaborado por: El Investigador

<b>Actor 02:</b>	Secretaria
<b>Descripción:</b>	Se encarga de generar los reportes de asistencia, informes de las horas laboradas y anomalías en las jornadas laborales.

Tabla 7. Descripción del Actor Secretaria. Elaborado por: El Investigador

#### 4.2.1.2. Descripción de los Casos de Uso

Para describir los casos de uso se debe tomar en cuenta los siguientes ítems:

- Actor: El actor que interactúa en el caso de uso.
- Precondiciones: Los requisitos necesarios para llevar a cabo el caso de uso.
- Descripción: Una breve descripción de lo que realiza el caso de uso.
- Proceso: Pasos del proceso a seguir.

<b>Registrar ingreso a jornada laboral</b>	
<b>Actor:</b>	Colaborador
<b>Precondiciones:</b>	El colaborador debe dirigirse al lugar de registro de asistencia
<b>Descripción:</b>	Proceso que permite verificar la presencia del colaborador en su lugar de trabajo.
<b>Proceso:</b>	<ul style="list-style-type: none"><li>• Ingresar credenciales.</li><li>• Validar credenciales.</li><li>• Verificar hora de registro.</li><li>• Almacenar registro.</li><li>• Enviar mensaje de confirmación o de error según sea el caso.</li></ul>

Tabla 8. Descripción Registrar hora de ingreso a jornada laboral. Elaborado por: El Investigador

<b>Registrar salida de jornada laboral</b>	
<b>Actor:</b>	Colaborador
<b>Precondiciones:</b>	El colaborador asistió a su lugar de trabajo.
<b>Descripción:</b>	Proceso que permite registrar la hora de salida del lugar de trabajo tras cumplir la jornada laboral.
<b>Proceso:</b>	<ul style="list-style-type: none"><li>• Ingresar credenciales.</li><li>• Validar credenciales.</li><li>• Verificar hora de registro.</li><li>• Almacenar registro.</li><li>• Enviar mensaje de confirmación o de error según sea el caso.</li></ul>

Tabla 9. Descripción de Registrar hora de salida de jornada laboral. Elaborado por: El Investigador

<b>Validar registros de asistencias</b>	
<b>Actor:</b>	Secretaria

<b>Descripción:</b>	Proceso que permite verificar el cumplimiento o incumplimiento en la asistencia a las jornadas laborales.
<b>Proceso:</b>	<ul style="list-style-type: none"> <li>• Verificar registro de asistencia.</li> <li>• Verificar hora de registro.</li> <li>• Comparar la hora de registro con la hora establecida en los horarios de trabajo.</li> <li>• Verificar atraso o salida anticipada a la hora establecida en los horarios de trabajo.</li> <li>• Registrar falta, atraso o salida anticipada.</li> </ul>

Tabla 10. Descripción de Registrar anomalías en la asistencia. Elaborado por: El Investigador

<b>Generar informe de asistencia</b>	
<b>Actor:</b>	Secretaria
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• Poseer los registros de ingresos y salidas de los colaboradores a las jornadas laborales.</li> </ul>
<b>Descripción:</b>	Proceso que permite crear un informe donde se registra las horas de ingreso y salida a las jornadas laborales de los colaboradores, en el mismo se registra si fue un día normal de asistencia o si existieron anomalías.
<b>Proceso:</b>	<ul style="list-style-type: none"> <li>• Verificar los registros de asistencia del día.</li> <li>• Verificar las horas de registro.</li> <li>• Registrar atrasos en caso de existir.</li> <li>• Registrar salida anticipada en caso de existir.</li> <li>• Generar informe con la información de registros, faltas, atrasos y/o salidas anticipadas.</li> </ul>

Tabla 11. Descripción de Generar informe de asistencia. Elaborado por: El Investigador

<b>Generar reporte de días laborados por mes</b>	
<b>Actor:</b>	Secretaria
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• Poseer los informes de asistencia diaria.</li> </ul>
<b>Descripción:</b>	Proceso que permite registrar el número de días laborados en el mes.
<b>Proceso:</b>	<ul style="list-style-type: none"> <li>• Verificar los registros de asistencia diarias.</li> <li>• Calcular el número de días registrados.</li> <li>• Generar reporte con el número de días laborados</li> </ul>

Tabla 12. Descripción de Generar reporte de días laborados por mes. Elaborado por: El Investigador



<b>Generar reporte de asistencia por mes</b>	
<b>Actor:</b>	Secretaria
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• Poseer los informes de asistencia.</li> </ul>
<b>Descripción:</b>	Proceso que permite generar el reporte de asistencia del colaborador en el mes.
<b>Proceso:</b>	<ul style="list-style-type: none"> <li>• Verificar el informe de asistencia</li> <li>• Verificar el informe de días laborados en el mes</li> <li>• Generar reporte con la información registrada en los informes de asistencia, el número de días laborados, y los registros de asistencia.</li> </ul>

*Tabla 13. Descripción de Generar reporte de asistencia por mes. Elaborado por: El Investigador*

#### **4.2.1.3. Diagramas de Secuencia**

“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.” [22]

Con estos diagramas se entenderá de mejor manera los procesos que el sistema realizará, ya que se diagrama a detalle por que se realiza el proceso de cada caso de uso paso a paso como se muestran en las imágenes 11, 12, 13, 14, 15, 16.

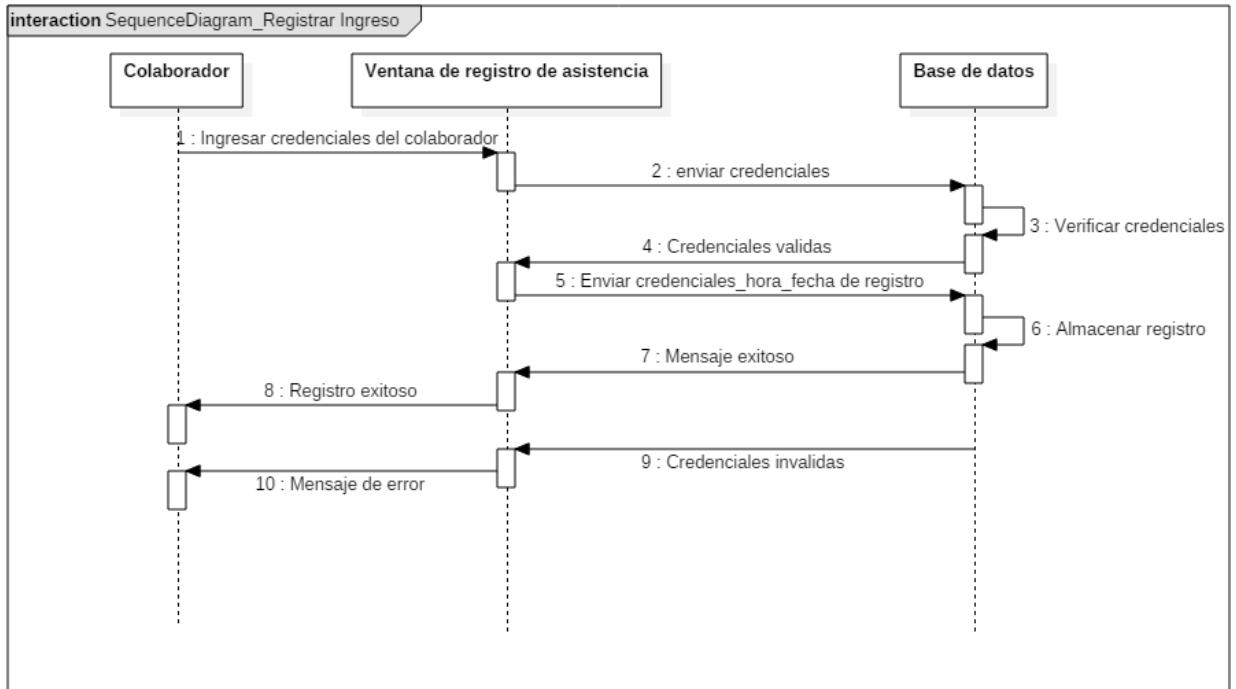


Imagen 11. Diagrama de secuencia de registro de ingresos. Elaborado por el investigador.

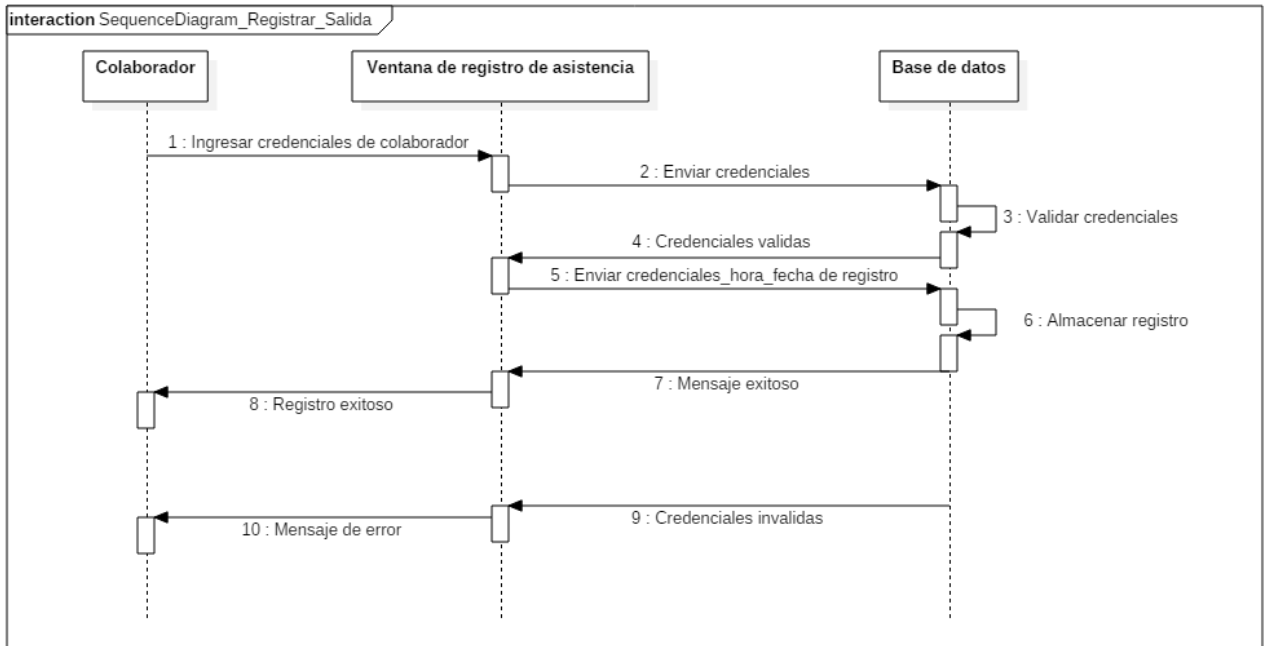


Imagen 12. Diagrama de secuencia de registro de salida. Elaborado por el investigador.

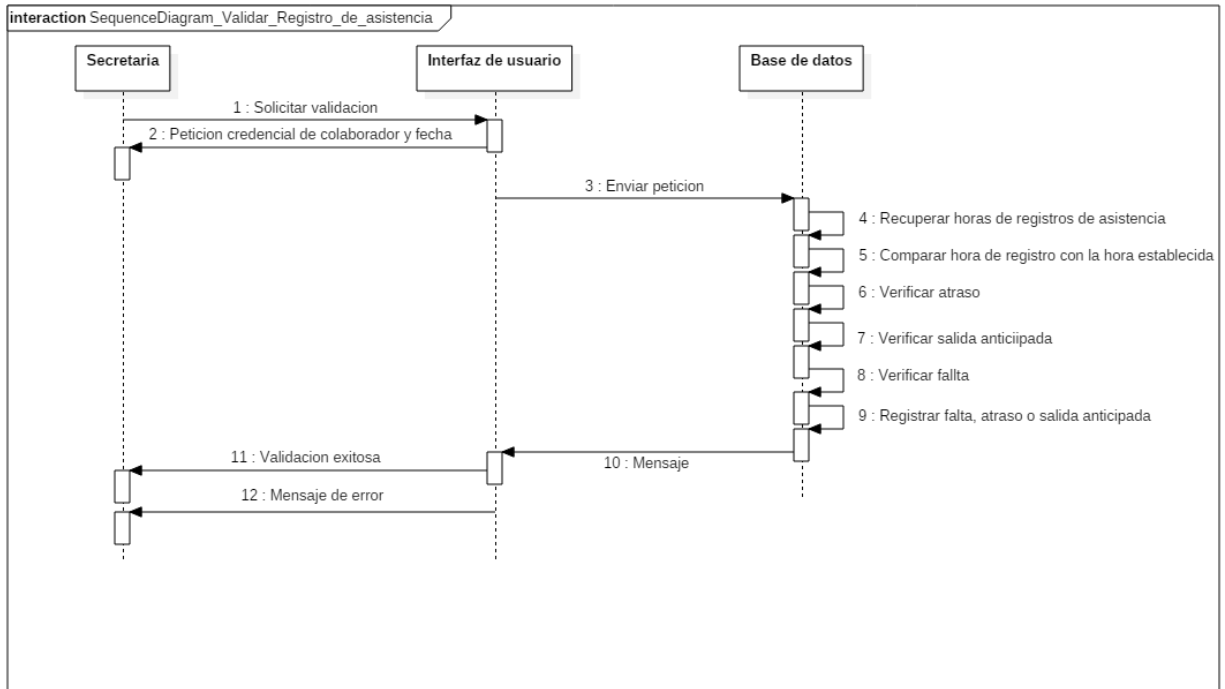


Imagen 13. Diagrama de secuencia de Validar registros de asistencia. Elaborado por el investigador

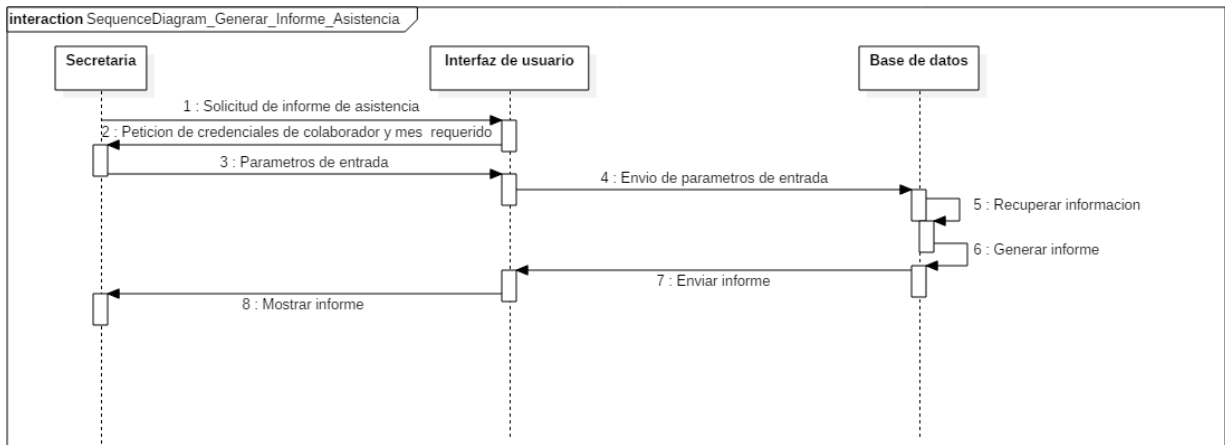


Imagen 14. Diagrama de secuencia de generación de informes de asistencia. Elaborado por el investigador.

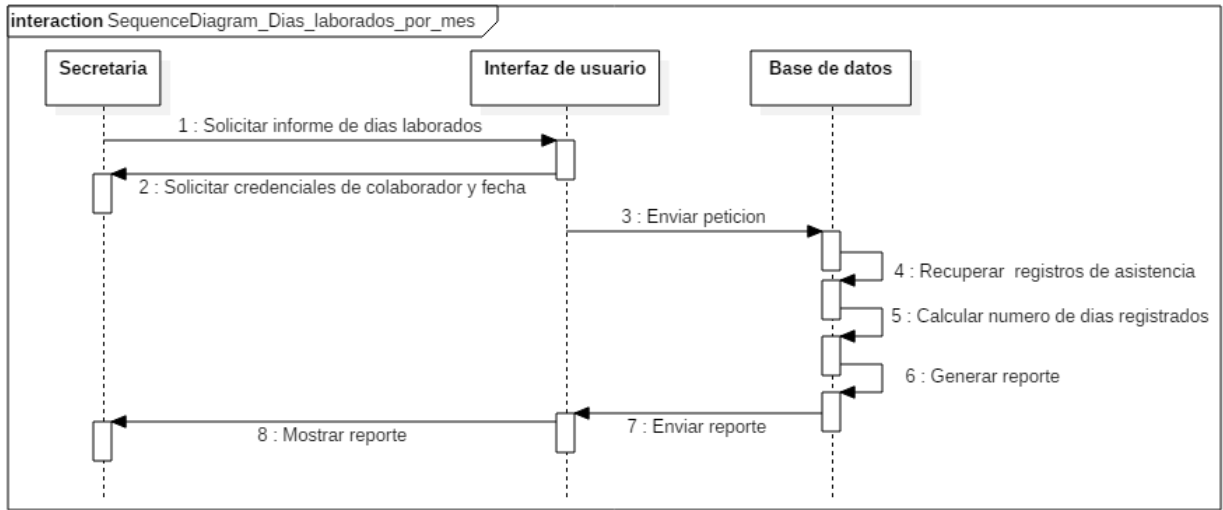


Imagen 15. Diagrama de secuencia de Generar reporte de días laborados en el mes. Elaborado por el investigador

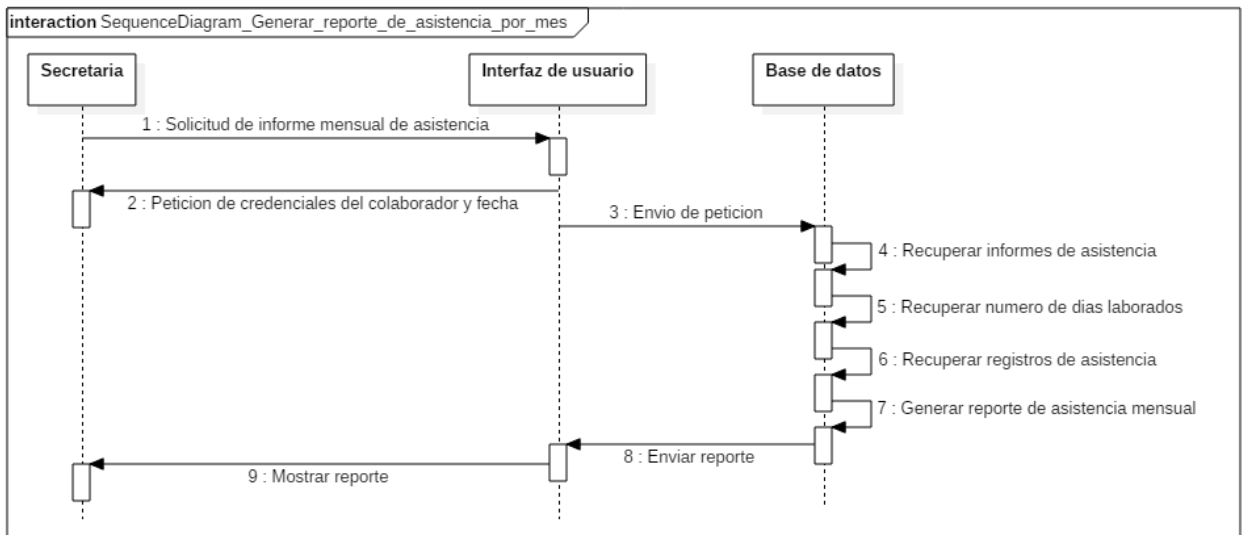


Imagen 16. Diagrama de secuencia de Generar reporte de asistencia mensual. Elaborado por el Investigador

## 4.2.2. Diseño de Base de Datos

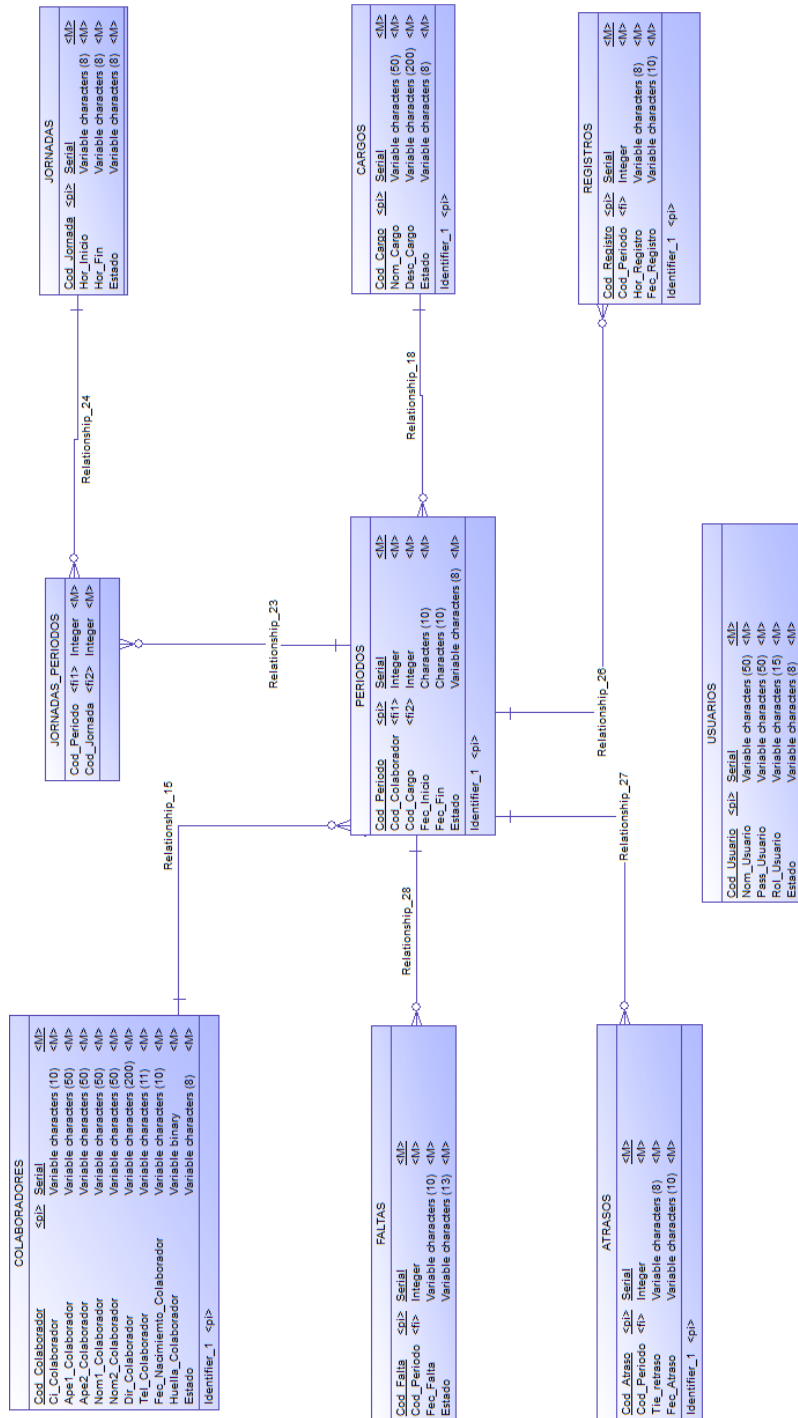


Imagen 17. Diseño Base de Datos. Elaborado por: El Investigado

#### 4.2.2.1. Diccionario de Datos

Atrasos				
Campo	Tipo	Longitud	Descripción	Restricciones
cod_atraso	Integer		Código secuencial de la tabla de atrasos	Clave primaria
cod_periodo	Integer		Código de la tabla periodo	Clave foránea
tie_retraso	character varying	8	Tiempo de atraso a la jornada	Obligatorio
fec_atraso	character varying	10	Fecha en que se registra el atraso	Obligatorio

Tabla 14. Diccionario de datos de la tabla "Atrasos". Elaborado por el investigador

Colaboradores				
Campo	Tipo	Longitud	Descripción	Restricciones
cod_colaborador	integer		Código secuencial de la tabla de Colaboradores	Clave primaria
ci_colaborador	character varying	10	Cedula del colaborador	Obligatorio
ape1_colaborador	character varying	50	Primer apellido	Obligatorio
ape2_colaborador	character varying	50	Segundo apellido	Obligatorio
nom1_colaborador	character varying	50	Primer nombre	Obligatorio
nom2_colaborador	character varying	50	Segundo nombre	Obligatorio
dir_colaborador	character varying	200	Dirección del colaborador	Obligatorio
tel_colaborador	character varying	11	Teléfono del colaborador	Obligatorio
fec_nacimiento_colaborador	character varying	10	Fecha de nacimiento del colaborador	Obligatorio
huella_colaborador	bytea		Huella dactilar del colaborador	Obligatorio
estado	character varying	8	Puede ser "Activo" o "Inactivo"	Obligatorio

Tabla 15. Diccionario de datos de la tabla "Colaboradores". Elaborado por el investigador

Cargos				
Campo	Tipo	Longitud	Descripción	Restricciones
cod_cargo	integer		Código secuencial de la tabla de cargos	Clave primaria
nom_cargo	character varying	50	Nombre del cargo	Obligatorio
desc_cargo	character varying	200	Descripción del cargo	Obligatorio
estado	character varying	8	Puede ser "Activo" o "Inactivo"	Obligatorio

Tabla 16. Diccionario de datos de la tabla "Cargos". Elaborado por el investigador

Faltas				
Campo	Tipo	Longitud	Descripción	Restricciones
cod_falta	integer		Código secuencial de la tabla de Faltas	Clave primaria
cod_periodo	integer		Código de la tabla periodos	Clave foranea
fec_falta	character varying	10	Fecha en que se registra la falta	Obligatorio
estado	character varying	13	Puede ser "Activo" o "Inactivo"	Obligatorio

Tabla 17. Diccionario de datos de la tabla "Faltas". Elaborado por el investigador

Jornadas				
Campo	Tipo	Longitud	Descripción	Restricciones
cod_jornada	integer		Código secuencial de la tabla de Jornadas	Clave primaria
hor_inicio	character varying	8	Hora de inicio de la jornada	Obligatorio
hor_fin	character varying	8	Hora de fin de la jornada	Obligatorio
estado	character varying	8	Puede ser "Activo" o "Inactivo"	Obligatorio

Tabla 18. Diccionario de datos de la tabla "Jornadas". Elaborado por el investigador

Jornadas_Periodos				
Campo	Tipo	Longitud	Descripción	Restricciones
cod_periodo	integer		Código de la tabla de Periodos	Clave foranea
cod_jornada	integer		Código de la tabla Jornadas	Clave foranea

Tabla 19. Diccionario de datos de la tabla "Jornadas\_Periodos". Elaborado por el investigador

Periodos				
Campo	Tipo	Longitud	Descripción	Restricciones
cod_periodo	integer		Código secuencial de la tabla de Periodos	Clave primaria
cod_colaborador	integer		Código de la tabla Colaboradores	Clave foranea
cod_cargo	integer		Código de la tabla Cargos	Clave foranea
fec_inicio	character	10	Fecha de inicio del periodo	Obligatorio
fec_fin	character	10	Fecha del fin del periodo	Obligatorio
estado	character varying	8	Puede ser "Activo" o "Inactivo"	Obligatorio

Tabla 20. Diccionario de datos de la tabla "Periodos". Elaborado por el investigador

Registros				
Campo	Tipo	Longitud	Descripción	Restricciones
cod_registro	integer		Código secuencial de la tabla de Registros	Clave primaria
cod_periodo	integer		Código de la tabla periodos	Clave foranea
hor_registro	character varying	8	Hora del registro	Obligatorio
fec_registro	character varying	10	Fecha del registro	Obligatorio

Tabla 21. Diccionario de datos de la tabla "Registros". Elaborado por el investigador

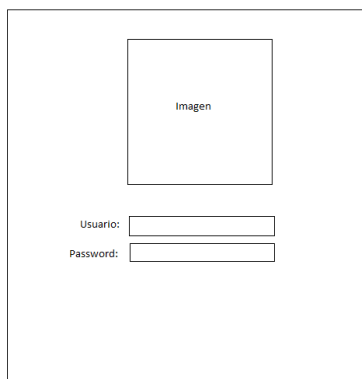
Usuarios				
Campo	Tipo	Longitud	Descripción	Restricciones
cod_usuario	integer		Código secuencial de la tabla de Usuarios	Clave primaria
nom_usuario	character varying	50	Nombre de usuario	Obligatorio
pass_usuario	character varying	50	Password	Obligatorio
rol_usuario	character varying	15	Puede ser "Normal" o "Admin"	Obligatorio

Tabla 22. Diccionario de datos de la tabla "Usuarios". Elaborado por el investigador



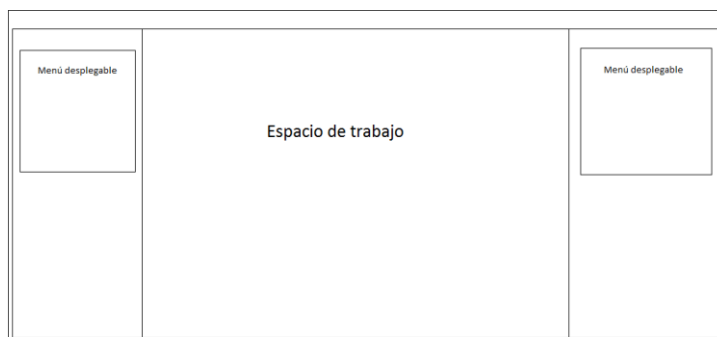
### 4.2.3. Diseño de la Interfaz de Usuario

**Inicio de Sesión:** Permite al usuario el ingreso al sistema de bienes tecnologicos, previamente se realiza una validacion de usuario y contraseña con la base de datos, de esta manera podra acceder a los modulos en donde se tenga permisos.



*Imagen 18. Interfaz de login. Elaborado por el investigador.*

**Página principal:** Permite acceder a todas las funcionalidades que posee la aplicación web.



*Imagen 19. Interfaz de la página administradora de las funcionalidades del software. Elaborado por el investigador.*

**Crear, leer, actualizar, borrar (C.R.U.D.):** Estructura genérica de las interfaces para la manipulación de la información en la base de datos.

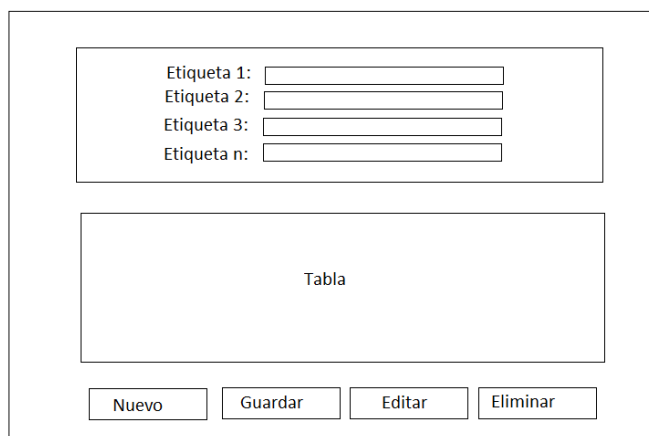


Imagen 20. Estructura genérica de las interfaces para manipular la información de la base de datos. Elaborado por el Investigador.

## Implementación de la tecnología GWT

### 4.3. Codificación de la Aplicación Web

La codificación de las aplicaciones web se divide en 2, el lado del cliente que es toda interfaz de usuario que va a manipular el usuario y el lado del servidor en donde se realiza validaciones de datos, envío de datos para la presentación de información, recibo de datos para el ingreso y actualización de información en la base de datos y en general para la manipulación de la información.

#### 4.3.1. Codificación del Lado del Cliente

- **Login:** página que se encarga de validar los usuarios que tienen acceso a las funcionalidades del sistema.

```

package w.i.s.p.client;

import com.extjs.gxt.ui.client.widget.Window;
import com.extjs.gxt.ui.client.widget.layout.BorderLayout;
import com.extjs.gxt.ui.client.widget.form.FormPanel;
import com.extjs.gxt.ui.client.widget.layout.BorderLayoutData;
import com.extjs.gxt.ui.client.Style.LayoutRegion;
import com.extjs.gxt.ui.client.widget.layout.AbsoluteLayout;
import com.google.gwt.user.client.rpc.AsyncCallback;
import com.google.gwt.user.client.ui.Image;
import com.extjs.gxt.ui.client.widget.layout.AbsoluteData;
import com.extjs.gxt.ui.client.Style;
import com.extjs.gxt.ui.client.widget.Text;
import com.extjs.gxt.ui.client.widget.form.TextField;
import com.google.gwt.user.client.ui.PasswordTextBox;

```

```

import com.google.gwt.user.client.ui.RootPanel;
import com.extjs.gxt.ui.client.widget.button.Button;
import com.extjs.gxt.ui.client.Style.HorizontalAlignment;
import com.extjs.gxt.ui.client.event.SelectionListener;
import com.extjs.gxt.ui.client.event.ButtonEvent;
import com.extjs.gxt.ui.client.event.Events;
import com.extjs.gxt.ui.client.event.Listener;
import w.i.s.p.server.*;
public class Login extends Window
{
    private PasswordTextBox txtPassword;
    private TextField txtUsuario;
    public Login()
    {
        setButtonAlign(HorizontalAlignment.CENTER);
        setResizable(false);
        setSize("429px", "480px");
        setClosable(false);
        setLayout(new BorderLayout());
        FormPanel frmPnlLogin = new FormPanel();
        frmPnlLogin.setHeading("Login");
        frmPnlLogin.setCollapsible(true);
        frmPnlLogin.setLayout(new AbsoluteLayout());
        Image image = new Image("Imagenes/login.png");
        frmPnlLogin.add(image, new AbsoluteData(108, 6));
        image.setSize("201px", "181px");

        Text lblUsuario = new Text("<label>Usuario:</label>");
        frmPnlLogin.add(lblUsuario, new AbsoluteData(75, 215));

        Text txtContrasena = new Text("<label>Contrase\u00F1a:</label>");
        frmPnlLogin.add(txtContrasena, new AbsoluteData(53, 257));

        txtUsuario = new TextField();
        frmPnlLogin.add(txtUsuario, new AbsoluteData(142, 209));
        txtUsuario.setSize("150px", "25px");

        txtPassword = new PasswordTextBox();
        frmPnlLogin.add(txtPassword, new AbsoluteData(142, 248));
        txtPassword.setSize("140px", "13px");

        Button btnIngresar = new Button("<buton>Ingresar</buton>");
        btnIngresar.addSelectionListener(new SelectionListener<ButtonEvent>() {
            public void componentSelected(ButtonEvent ce)
            {
                login();
            }
        });
        frmPnlLogin.add(btnIngresar, new AbsoluteData(110, 306));
        btnIngresar.setSize("90px", "34px");

        Button btnCancelar = new Button("<buton>Cancelar</buton>");
        frmPnlLogin.add(btnCancelar, new AbsoluteData(219, 306));
        btnCancelar.setSize("90px", "34px");
        add(frmPnlLogin, new BorderLayoutData(LayoutRegion.CENTER));
    }

    private void login(){

```

```

        RcpServicio.Util.getInstance().InicioSesion(txtUsuario.getValue().toString(),
        txtPassword.getText(), new AsyncCallback<String>() {

            @Override
            public void onFailure(Throwable caught) {
                // TODO Auto-generated method stub
                com.google.gwt.user.client.Window.alert("Algo salio
mal");
            }

            @Override
            public void onSuccess(String result) {
                // TODO Auto-generated method stub
                if(result.equals("Se conecto a la base"))
                {
                    com.google.gwt.user.client.Window.alert("Registro
exitoso");

                    RootPanel panel= RootPanel.get();
                    //Login obj= new Login();
                    //Registro ventana = new Registro();
                    Admin ventana= new Admin();
                    panel.add(ventana);
                    ventana.center();
                }
                else
                {
                    com.google.gwt.user.client.Window.alert(result+",
credenciales no existentes");
                }
            }
        });
    }
}

```

- **Página principal:** página que permite acceder a todas las funcionalidades de la aplicación web.

```

package w.i.s.p.client;

import com.extjs.gxt.ui.client.widget.Window;
import com.extjs.gxt.ui.client.widget.layout.BorderLayout;
import com.extjs.gxt.ui.client.widget.grid.ColumnConfig;
import com.extjs.gxt.ui.client.widget.tree.Tree;
import com.extjs.gxt.ui.client.widget.layout.BorderLayoutData;
import com.extjs.gxt.ui.client.Style.LayoutRegion;
import com.extjs.gxt.ui.client.widget.tree.TreeItem;

import w.i.s.p.shared.Registro;

import com.extjs.gxt.ui.client.widget.form.FormPanel;
import com.extjs.gxt.ui.client.widget.VerticalPanel;

```

```

import com.extjs.gxt.ui.client.widget.button.Button;
import com.extjs.gxt.ui.client.widget.layout.FormData;
import com.extjs.gxt.ui.client.widget.button.IconButton;
import com.extjs.gxt.ui.client.widget.ContentPanel;
import com.extjs.gxt.ui.client.widget.TabItem;
import com.extjs.gxt.ui.client.widget.layout.RowLayout;
import com.extjs.gxt.ui.client.Style.Orientation;
import com.extjs.gxt.ui.client.widget.layout.RowData;
import com.extjs.gxt.ui.client.Style;
import com.extjs.gxt.ui.client.util.Margins;
import com.extjs.gxt.ui.client.event.SelectionListener;
import com.extjs.gxt.ui.client.event.ButtonEvent;
import com.extjs.gxt.ui.client.event.Events;
import com.extjs.gxt.ui.client.event.Listener;
import com.extjs.gxt.ui.client.widget.TabPanel;
import com.google.gwt.user.client.ui.Label;
import com.google.gwt.user.client.ui.Image;

public class Admin extends Window
{
    private TabPanel tabPanel;
    protected Aportaciones aportaciones;
    protected Cargos cargos;
    protected Colaboradores colaboradores;
    protected Egresos egresos;
    protected GenerarRol generarRol;
    protected HorasExtras horasExtras;
    protected Ingresos ingresos;
    protected Jornadas jornadas;
    protected RolConfiguracion rolConfiguracion;
    protected Provisiones provisiones;
    protected RegistroAsistencia registroAsistencia;
    protected Usuarios usuarios;

    public Admin() {
        setMaximizable(true);
        setResizable(false);
        setSize("1200px", "550px");
        setLayout(new BorderLayout());

        VerticalPanel verticalPanel = new VerticalPanel();

        FormPanel formPersonal = new FormPanel();
        formPersonal.setHeading("Personal");
        formPersonal.setCollapsible(true);

        Button btnColaboradores = new Button("Colaboradores");
        btnColaboradores.addListener(Events.Select, new Listener<ButtonEvent>()
        {
            public void handleEvent(ButtonEvent e)
            {
                colaboradores= new Colaboradores();
                Mostrar(colaboradores , "Colaboradores");
            }
        });
        btnColaboradores.addSelectionListener(new
        SelectionListener<ButtonEvent>() {
            public void componentSelected(ButtonEvent ce)
            {

```

```

    }
});
formPersonal.add(btnColaboradores, new FormData("100%"));

Button btnCargos = new Button("Cargos");
btnCargos.addListener(Events.Select, new Listener<ButtonEvent>() {
    public void handleEvent(ButtonEvent e)
    {
        cargos= new Cargos();
        Mostrar(cargos, "Cargos");
    }
});
formPersonal.add(btnCargos, new FormData("100%"));

Button btnJornadas = new Button("Jornadas");
btnJornadas.addListener(Events.Select, new Listener<ButtonEvent>() {
    public void handleEvent(ButtonEvent e)
    {
        jornadas= new Jornadas();
        Mostrar(jornadas, "Jornadas");
    }
});
formPersonal.add(btnJornadas, new FormData("100%"));

Button btnUsuarios = new Button("Usuarios");
btnUsuarios.addListener(Events.Select, new Listener<ButtonEvent>() {
    public void handleEvent(ButtonEvent e)
    {
        usuarios= new Usuarios();
        Mostrar(usuarios, "Usuarios");
    }
});
formPersonal.add(btnUsuarios, new FormData("100%"));
verticalPanel.add(formPersonal);
formPersonal.setSize("173px", "142px");

FormPanel formSalarios = new FormPanel();
formSalarios.setHeading("Salarios");
formSalarios.setCollapsible(true);

Button btnAportaciones = new Button("Aportaciones");
btnAportaciones.addListener(Events.Select, new Listener<ButtonEvent>() {
    public void handleEvent(ButtonEvent e)
    {
        aportaciones= new Aportaciones();
        Mostrar(aportaciones, "Aportaciones");
    }
});
formSalarios.add(btnAportaciones, new FormData("100%"));

Button btnEgresos = new Button("Egresos");
btnEgresos.addListener(Events.Select, new Listener<ButtonEvent>() {
    public void handleEvent(ButtonEvent e)
    {
        egresos= new Egresos();
        Mostrar(egresos, "Egresos");
    }
});

```

```

formSalarios.add(btnEgresos, new FormData("100%"));
btnEgresos.setSize("151", "23");

Button btnIngresos = new Button("Ingresos");
btnIngresos.addListener(Events.Select, new Listener<ButtonEvent>() {
    public void handleEvent(ButtonEvent e)
    {
        ingresos = new Ingresos();
        Mostrar(ingresos,"Ingresos");
    }
});
formSalarios.add(btnIngresos, new FormData("100%"));

Button btnProvisiones = new Button("Provisiones");
btnProvisiones.addListener(Events.Select, new Listener<ButtonEvent>() {
    public void handleEvent(ButtonEvent e)
    {
        provisiones = new Provisiones();
        Mostrar(provisiones,"Provisiones");
    }
});
formSalarios.add(btnProvisiones, new FormData("100%"));

Button btnHorasExtras = new Button("Horas extras");
btnHorasExtras.addListener(Events.Select, new Listener<ButtonEvent>() {
    public void handleEvent(ButtonEvent e)
    {
        horasExtras= new HorasExtras();
        Mostrar(horasExtras, "Horas extras");
    }
});
formSalarios.add(btnHorasExtras, new FormData("100%"));
verticalPanel.add(formSalarios);
formSalarios.setSize("173px", "170px");

Image image = new Image("Imagenes/AV-GIRL.JPG");
verticalPanel.add(image);
image.setSize("173px", "63px");
add(verticalPanel, new BorderLayoutData(LayoutRegion.WEST, 175.0f));

VerticalPanel verticalPanel_1 = new VerticalPanel();

FormPanel formAsistencia = new FormPanel();
formAsistencia.setHeading("Asistencia");
formAsistencia.setCollapsible(true);

Button btnVer = new Button("Ver");
formAsistencia.add(btnVer, new FormData("100%"));

Button btnJustificar = new Button("Justificar");
formAsistencia.add(btnJustificar, new FormData("100%"));

Button btnInformes = new Button("Informes");
formAsistencia.add(btnInformes, new FormData("100%"));
verticalPanel_1.add(formAsistencia);
formAsistencia.setSize("173px", "142px");

FormPanel formRoles = new FormPanel();
formRoles.setHeading("Roles");

```

```

formRoles.setCollapsible(true);

Button btnVer_1 = new Button("Ver");
formRoles.add(btnVer_1, new FormData("100%"));

Button btnGenerar = new Button("Generar");
btnGenerar.addListener(Events.Select, new Listener<ButtonEvent>() {
    public void handleEvent(ButtonEvent e)
    {
        generarRol = new GenerarRol();
        Mostrar(generarRol,"Generar rol de pagos");
    }
});
formRoles.add(btnGenerar, new FormData("100%"));

Button btnConfigurar = new Button("Configurar");
btnConfigurar.addListener(Events.Select, new Listener<ButtonEvent>() {
    public void handleEvent(ButtonEvent e)
    {
        rolConfiguracion= new RolConfiguracion();
        Mostrar(rolConfiguracion,"Configuracion de rol de
pagos");
    }
});
formRoles.add(btnConfigurar, new FormData("100%"));
verticalPanel_1.add(formRoles);
formRoles.setSize("173px", "142px");

Image image_1 = new Image("Imágenes/gmb.JPG");
verticalPanel_1.add(image_1);
image_1.setSize("186px", "101px");
add(verticalPanel_1, new BorderLayoutData(LayoutRegion.EAST, 190.0f));

tabPanel = new TabPanel();
tabPanel.setTabScroll(true);
add(tabPanel, new BorderLayoutData(LayoutRegion.CENTER));
}
public void Mostrar(TabItem ventana,String label)
{
    ventana.setClosable(true);
    ventana.setText(label);
    tabPanel.add(ventana);
    ventana.setVisible(true);
}
}
}

```



- **C.R.U.D.:** Pagina encargada de ingresar, editar, eliminar información en la base de datos.

```

package w.i.s.p.client;

import com.extjs.gxt.ui.client.widget.TabItem;
import com.extjs.gxt.ui.client.widget.layout.AbsoluteLayout;
import com.extjs.gxt.ui.client.widget.form.LabelField;
import com.extjs.gxt.ui.client.widget.layout.AbsoluteData;
import com.extjs.gxt.ui.client.Style;
import com.extjs.gxt.ui.client.event.Events;
import com.extjs.gxt.ui.client.event.FieldEvent;
import com.extjs.gxt.ui.client.event.Listener;
import com.extjs.gxt.ui.client.widget.form.TextField;
import com.extjs.gxt.ui.client.widget.form.TimeField;
import com.extjs.gxt.ui.client.widget.button.Button;
import com.extjs.gxt.ui.client.widget.table.Table;
import com.extjs.gxt.ui.client.widget.table.TableColumnModel;
import com.extjs.gxt.ui.client.widget.table.TableColumn;
import com.google.gwt.user.client.ui.ScrollPanel;
import com.google.gwt.user.cellview.client.CellTable;
import com.google.gwt.user.cellview.client.TextColumn;
import com.google.gwt.user.cellview.client.Column;

import java.sql.Time;
import java.util.Date;
import java.util.List;

import com.google.gwt.cell.client.DatePickerCell;
import com.google.gwt.dom.client.BrowserEvents;
import com.google.gwt.cell.client.DateCell;
import com.google.gwt.view.client.CellPreviewEvent;
import com.google.gwt.view.client.SingleSelectionModel;
import com.google.gwt.view.client.CellPreviewEvent.Handler;
import com.google.gwt.event.dom.client.KeyPressEvent;
import com.google.gwt.event.dom.client.KeyPressHandler;
import com.google.gwt.user.client.Window;
import com.google.gwt.user.client.rpc.AsyncCallback;
import com.google.gwt.view.client.CellPreviewEvent;
import com.extjs.gxt.ui.client.event.ButtonEvent;

public class Jornadas extends TabItem
{
    private w.i.s.p.shared.Jornadas jornadasGlobal;
    private TimeField timeInicio;
    private TimeField timeFin;
    private Button btnNuevo;
    private Button btnGuardar;
    private Button btnEditar;
    private Button btnEliminar;
    private TextField txtBuscar;
    private LabelField lblBuscar;
    private String proceso;
    private CellTable<w.i.s.p.shared.Jornadas> dataGridDatos;

```

```

private TextColumn<w.i.s.p.shared.Jornadas> textColumn;
private TextColumn<w.i.s.p.shared.Jornadas> textColumn_1;
public Jornadas() {
    proceso="";
    jornadasGlobal=new w.i.s.p.shared.Jornadas();
    setLayout(new AbsoluteLayout());

    LabelField lblfldHoraDeInicio = new LabelField("<label>Hora de
inicio:</label>");
    add(lblfldHoraDeInicio, new AbsoluteData(381, 82));

    LabelField lblfldHoraDeFinalizacin = new LabelField("<label>Hora de
Finalizaci\u00F3n:</label>");
    add(lblfldHoraDeFinalizacin, new AbsoluteData(339, 120));

    timeInicio = new TimeField();
    timeInicio.setIncrement(1);
    add(timeInicio, new AbsoluteData(480, 81));
    timeInicio.setSize("67px", "22px");

    timeFin = new TimeField();
    timeFin.setIncrement(1);
    add(timeFin, new AbsoluteData(480, 120));
    timeFin.setSize("67px", "22px");

    btnNuevo = new Button("<buton>Nuevo</buton>");
    btnNuevo.addListener(Events.Select, new Listener<ButtonEvent>() {
        public void handleEvent(ButtonEvent e)
        {
            Nuevo();
        }
    });
    add(btnNuevo, new AbsoluteData(277, 398));
    btnNuevo.setSize("100 px", "25 px");

    btnGuardar = new Button("<buton>Guardar</buton>");
    btnGuardar.addListener(Events.Select, new Listener<ButtonEvent>() {
        public void handleEvent(ButtonEvent e)
        {
            if(proceso!="editar")
                Insert();
            else
                Update();
        }
    });
    add(btnGuardar, new AbsoluteData(383, 398));
    btnGuardar.setSize("100 px", "25 px");

    btnEditar = new Button("<buton>Editar</buton>");
    btnEditar.addListener(Events.Select, new Listener<ButtonEvent>() {
        public void handleEvent(ButtonEvent e)
        {
            Editar();
        }
    });
    add(btnEditar, new AbsoluteData(489, 398));
    btnEditar.setSize("100 px", "25 px");

```

```

btnEliminar = new Button("<buton>Eliminar</buton>");
btnEliminar.addListener(Events.Select, new Listener<ButtonEvent>() {
    public void handleEvent(ButtonEvent e)
    {
        Delete();
    }
});
add(btnEliminar, new AbsoluteData(595, 398));
btnEliminar.setSize("100 px", "25 px");

ScrollPane scrollPanel = new ScrollPanel();
add(scrollPanel, new AbsoluteData(275, 201));
scrollPanel.setSize("420px", "170px");

dataGridDatos = new CellTable<w.i.s.p.shared.Jornadas>();
scrollPanel.setWidget(dataGridDatos);
dataGridDatos.setSize("100%", "100%");

textColumn = new TextColumn<w.i.s.p.shared.Jornadas>() {
    @Override
    public String getValue(w.i.s.p.shared.Jornadas object) {
        return object.getHor_Inicio();
    }
};
dataGridDatos.addColumn(textColumn, "Hora de inicio");

textColumn_1 = new TextColumn<w.i.s.p.shared.Jornadas>() {
    @Override
    public String getValue(w.i.s.p.shared.Jornadas object) {
        return object.getHor_Fin();
    }
};
dataGridDatos.addColumn(textColumn_1, "Hora de fin ");

SingleSelectionModel<w.i.s.p.shared.Jornadas> selectionModel = new
SingleSelectionModel<w.i.s.p.shared.Jornadas>();
dataGridDatos.setSelectionModel(selectionModel);

dataGridDatos.addCellPreviewHandler(new
Handler<w.i.s.p.shared.Jornadas>(){
    @Override
    public void
onCellPreview(CellPreviewEvent<w.i.s.p.shared.Jornadas> event) {
        // TODO Auto-generated method stub
        if
(BrowserEvents.CLICK.equals(event.getNativeEvent().getType())) {
            //it handles
            if(proceso=="editar")
            {
                jornadasGlobal = (w.i.s.p.shared.Jornadas)
((SingleSelectionModel)dataGridDatos.getSelectionModel()).getSelectedObject();
                CargarCampos();
            }
        }
    }
});
txtBuscar = new TextField();
txtBuscar.addListener(Events.KeyPress, new Listener<FieldEvent>() {

```

```

        public void handleEvent(FieldEvent e)
        {
            if(txtBuscar.getValue()==null)
            {
                CargarTabla("");
            }
            else
            {
                CargarTabla(txtBuscar.getValue().toString());
            }
        }
    });
    add(txtBuscar, new AbsoluteData(480, 161));

    lblBuscar = new LabelField("Buscar:");
    add(lblBuscar, new AbsoluteData(431, 162));
    BloqueoInicial();
    CargarTabla("");
}
//Metodos para controlar la secuencia de bloqueos de los botones segun las
//acciones a realiza en la base de datos
public void BloqueoInicial()
{
    timeInicio.setEnabled(false);
    timeFin.setEnabled(false);
    btnGuardar.setEnabled(false);
    btnEliminar.setEnabled(false);
    btnNuevo.setEnabled(true);
    btnEditar.setEnabled(true);
    lblBuscar.setVisible(false);
    txtBuscar.setVisible(false);
}

public void Nuevo()
{
    timeInicio.setEnabled(true);
    timeFin.setEnabled(true);
    btnNuevo.setEnabled(false);
    btnEditar.setEnabled(false);
    btnGuardar.setEnabled(true);
    btnEliminar.setEnabled(false);
    proceso="nuevo";
}

public void Editar()
{
    timeInicio.setEnabled(true);
    timeFin.setEnabled(true);
    btnNuevo.setEnabled(false);
    btnEditar.setEnabled(false);
    btnGuardar.setEnabled(true);
    btnEliminar.setEnabled(true);
    proceso="editar";
    lblBuscar.setVisible(true);
    txtBuscar.setVisible(true);
}

public void Guardar()

```

```

        {
            BloqueoInicial();
            proceso="guardar";
        }
        public void Eliminar()
        {
            BloqueoInicial();
            proceso="eliminar";
        }
        //Metodo para cargar la tabla de datos en la interfaz
        public void CargarTabla(String value)
        {
            RpcServiceJornadas.Util.getInstance().SelectInteligente(value,new
AsyncCallback<List<w.i.s.p.shared.Jornadas>>()
            {
                @Override
                public void onFailure(Throwable caught)
                {
                    // TODO Auto-generated method stub

                }

                @Override
                public void onSuccess(List<w.i.s.p.shared.Jornadas>
result)
                {
                    // TODO Auto-generated method stub
                    dataGridDatos.setRowData(result);
                    dataGridDatos.redraw();

                }

            });

        }
        //Metodo para cargar los campos con datos recogidos de la tabla
        public void CargarCampos()
        {
            if(timeInicio.isEnabled())
            {
                timeInicio.setRawValue(jornadasGlobal.getHor_Inicio());
                timeFin.setRawValue(jornadasGlobal.getHor_Fin());
            }
        }
        //Metodos de manipulacion de datos en la base de datos
        public void Insert()
        {
            if(ValidarDatos())
            {
                RpcServiceJornadas.Util.getInstance().ExisteJornada(jornadasGlobal,new
AsyncCallback<String>() {

                    @Override
                    public void onFailure(Throwable caught)
                    {
                        // TODO Auto-generated method stub

```

```

        Window.alert("error:
"+caught.getMessage()+"*****"+caught.getCause()+"*****");
        caught.printStackTrace();
    }

    @Override
    public void onSuccess(String result)
    {
        // TODO Auto-generated method stub
        if(result.equals("No"))
        {

            RpcServiceJornadas.Util.getInstance().Insert(jornadasGlobal, new
AsyncCallback<String>(){

                @Override
                public void

onFailure(Throwable caught)

                stub

                //com.google.gwt.user.client.Window.alert("Algo salio mal:
"+caught.getMessage());

                Window.alert("Jornada

                Limpiar();
                CargarTabla("");
            }

            @Override
            public void onSuccess(String

result)

            stub

            if(result=="Se inserto")
                Window.alert("Jornada

            else
                Window.alert("Algo salio

mal, no se pudo crear la nueva jornada: "+result);

            }

        });
        Guardar();

    }
    else
    {
        Window.alert("Esta jornada ya

existe, por favor revise su informacion");
    }
}

});

```

```

    }
}
public void Update()
{
    if(Validardatos())
    {
        RpcServiceJornadas.Util.getInstance().Update(jornadasGlobal, new
AsyncCallback<String>(){
@Override
public void
onFailure(Throwable caught)
{
// TODO Auto-generated method
stub
//com.google.gwt.user.client.Window.alert("Algo salio mal:
"+caught.getMessage());
Window.alert("Jornada
Limpiar();
CargarTabla("");
Guardar();
}
@Override
public void onSuccess(String
result)
{
// TODO Auto-generated method
if(result=="Se actualizo")
Window.alert("Jornada
actualizada.");
else
Window.alert("Algo salio
mal, no se pudo actualizar la informacion: "+result);
CargarTabla("");
}
});
}
}
public void Delete()
{
if(Window.confirm("¿ Esta seguro que desea eliminar esta jornada
? "))
{
RpcServiceJornadas.Util.getInstance().Delete(jornadasGlobal.getCod_Jornada()),ne
w AsyncCallback<String>()
{

```

```

        @Override
        public void onFailure(Throwable caught)
        {
            // TODO Auto-generated method stub
            Window.alert(caught.getMessage());
        }

        @Override
        public void onSuccess(String result)
        {
            // TODO Auto-generated method stub
            Window.alert("Jornada eliminado");
            CargarTabla("");
        }

    });

    Limpiar();
    Eliminar();
}

public boolean ValidarDatos()
{
    boolean result=true;
    String mensaje="";

    int hora=timeInicio.getValue().getHour();
    int minutos=timeInicio.getValue().getMinutes();

    int horaf=timeFin.getValue().getHour();
    int minutosf=timeFin.getValue().getMinutes();

    String hor,min;
    hor="";
    min="";
    if(hora<10)
        hor="0"+hora;
    else
        hor="" +hora;

    if(minutos<10)
        min="0"+minutos;
    else
        min="" +minutos;

    String horf,minf;
    horf="";
    minf="";

    if(horaf<10)
        horf="0"+horaf;
    else
        horf=horaf+"";

    if(minutosf<10)
        minf="0"+minutosf;
    else
        minf=minutosf+"";
}

```



```

        if(timeInicio.getValue()==null)
        {
            mensaje=mensaje+"-Por favor seleccione una hora de inicio
de jornada \n";
            result=false;
        }
        if((timeFin.getValue()==null))
        {
            mensaje=mensaje+"-Por favor seleccione un hora de fin de
jornada \n";
            result=false;
        }
        if((hora==horaf)&&(minutos==minutosf))
        {
            mensaje=mensaje+"-Las horas de inicio y fin de jornada no
pueden ser iguales \n";
            result=false;
        }
        if((hora==horaf)&&(minutosf<minutos))
        {
            mensaje=mensaje+"-La hora de fin de jornada no puede ser
menor que la hora de inicio, por favor revise su informacion \n";
            result=false;
        }
        if(horaf<hora)
        {
            mensaje=mensaje+"-La hora de fin de jornada no puede ser
menor que la hora de inicio, por favor revise su informacion \n";
            result=false;
        }
        String horaI="";
        String horaF="";
        if(result)
        {
            horaI=hor+":"+min;
            horaF=horf+":"+minf;

            jornadasGlobal.setHor_Inicio(horaI);
            jornadasGlobal.setHor_Fin(horaF);
        }
        else
        {
            Window.alert(mensaje);
        }
        return result;
    }
    public void Limpiar()
    {
        timeInicio.setRawValue("");
        timeFin.setRawValue("");
        txtBuscar.setValue("");
    }
}

```

- **Página web para la generación de un reporte.**

```

package w.i.s.p.client;

import com.extjs.gxt.ui.client.widget.TabItem;
import com.extjs.gxt.ui.client.widget.layout.AbsoluteLayout;
import com.extjs.gxt.ui.client.widget.form.LabelField;
import com.extjs.gxt.ui.client.widget.layout.AbsoluteData;
import com.extjs.gxt.ui.client.Style;
import com.extjs.gxt.ui.client.widget.form.TextField;
import com.extjs.gxt.ui.client.widget.table.Table;
import com.extjs.gxt.ui.client.widget.table.TableColumnModel;
import com.extjs.gxt.ui.client.widget.table.TableColumn;
import com.extjs.gxt.ui.client.widget.button.Button;
import com.extjs.gxt.ui.client.widget.grid.Grid;
import com.extjs.gxt.ui.client.store.ListStore;
import com.extjs.gxt.ui.client.widget.grid.ColumnModel;
import java.util.Collections;
import com.extjs.gxt.ui.client.widget.grid.ColumnConfig;

public class GenerarReporte extends TabItem
{
    public GenerarRol()
    {
        setLayout(new AbsoluteLayout());

        LabelField lblfldcdula = new LabelField("<label>C\u00E9dula:</label>");
        add(lblfldcdula, new AbsoluteData(105, 48));

        TextField txtCedula = new TextField();
        add(txtCedula, new AbsoluteData(163, 47));
        txtCedula.setSize("150", "25");

        LabelField lblfldHorasExtras = new LabelField("<label>Horas
Extras:</label>");
        add(lblfldHorasExtras, new AbsoluteData(69, 87));

        Table table = new Table(new TableColumnModel(new TableColumn("id.1",
"First", 0.3f), new TableColumn("id.2", "Second", 0.7f)));
        add(table, new AbsoluteData(163, 129));
        table.setSize("260px", "198px");

        Button btnBuscar = new Button("<buton>Buscar</buton>");
        add(btnBuscar, new AbsoluteData(321, 48));
        btnBuscar.setSize("100 px", "25 px");

        Button btnAgrupar = new Button("<buton>Agrupar</buton>");
        add(btnAgrupar, new AbsoluteData(162, 87));
        btnAgrupar.setSize("100 px", "25 px");

        Button btnGenerar = new Button("<buton>Generar</buton>");
        add(btnGenerar, new AbsoluteData(320, 87));
        btnGenerar.setSize("100px", "25 px");
    }
}

```

- **Interfaz para acceder a los métodos en el servidor:**

```

package w.i.s.p.client;

import java.util.List;

import com.google.gwt.core.client.GWT;
import com.google.gwt.user.client.rpc.RemoteService;
import com.google.gwt.user.client.rpc.RemoteServiceRelativePath;

import w.i.s.p.shared.Colaboradores;

@RemoteServiceRelativePath("RpcServiceColaborador")
public interface RpcServiceColaborador extends RemoteService
{
    /**
     * Utility class for simplifying access to the instance of async service.
     */
    public static class Util {
        private static RpcServiceColaboradorAsync instance;
        public static RpcServiceColaboradorAsync getInstance(){
            if (instance == null) {
                instance = GWT.create(RpcServiceColaborador.class);
            }
            return instance;
        }
    }
    public Colaboradores BuscarColaborador(String cedula) throws Exception;
    public List<Colaboradores> BuscarColaboradores() throws Exception;
    public String Insertar(Colaboradores colaborador) throws Exception;
    public String Update(Colaboradores colaborador) throws Exception;
    public boolean ExisteColaborador(String cedula) throws Exception;
    public String Delete(int codigo) throws Exception;
}

```

- **Codificación del Lado del Servidor**

En esta codificación del lado del servidor están los métodos más importantes, en si son de manipulación de la base de datos. A continuación, se muestran solo de los métodos de los comandos DML.

### Método de búsqueda de un colaborador:

```

package w.i.s.p.server;

import w.i.s.p.client.RpcServiceColaborador;
import w.i.s.p.shared.Colaboradores;
import w.i.s.p.shared.RolBusquedaColaborador;

import java.sql.Connection;
import java.sql.DriverManager;

```

```

import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import com.google.gwt.user.server.rpc.RemoteServiceServlet;

public class RpcServiceColaboradorImpl extends RemoteServiceServlet implements
RpcServiceColaborador
{

    private Connection ObtenerConexion() throws Exception
    {
        Class.forName("org.postgresql.Driver").newInstance();
        String url="jdbc:postgresql://127.0.0.1:5432/AsistenciaGWT";
        return DriverManager.getConnection(url,"postgres","admin");
    }

    public Colaboradores BuscarColaborador(String cedula) throws Exception
    {
        String sql="SELECT cod_colaborador, ci_colaborador, ape1_colaborador,
ape2_colaborador, "
                + "
nom1_colaborador,nom2_colaborador,dir_colaborador,tel_colaborador,
fec_nacimiento_colaborador,huella_colaborador,estado "+
                "FROM Colaboradores "+
                "WHERE ci_colaborador='"+cedula+"' "+
                "and estado='Activo'";
        Colaboradores obj=new Colaboradores();
        Connection cn=ObtenerConexion();
        Statement stmt;
        ResultSet rs;
        try
        {
            stmt = cn.createStatement();
            rs = stmt.executeQuery(sql);
            while ( rs.next() )
            {
                obj.setCod_Colaborador(rs.getInt("Cod_Colaborador"));
                obj.setCi_Colaborador(rs.getString("Ci_Colaborador"));
                obj.setApe1_Colaborador(rs.getString("Ape1_Colaborador"));
                obj.setApe2_Colaborador(rs.getString("Ape2_Colaborador"));
                obj.setNom1_Colaborador(rs.getString("Nom1_Colaborador"));
                obj.setNom2_Colaborador(rs.getString("Nom2_Colaborador"));
                obj.setDir_Colaborador(rs.getString("dir_colaborador"));
                obj.setTel_Colaborador(rs.getString("tel_colaborador"));

                obj.setFec_Nacimiento__Colaborador(rs.getString("fec_nacimiento_colaborador"));
                //obj.setHuella_Colaborador(rs.getBinaryStream("huella"));
                obj.setEstado(rs.getString("estado"));
            }

        }
        catch (SQLException e)
        {

```

```

        // TODO Auto-generated catch block
        System.out.print(e.getMessage());
    }

    finally
    {
        cn.close();
    }

    return obj;
}

public List<Colaboradores> BuscarColaboradores() throws Exception
{
    String sql="SELECT cod_colaborador, ci_colaborador, ape1_colaborador,
ape2_colaborador, nom1_colaborador,nom2_colaborador,dir_colaborador,tel_colaborador,
fec_nacimiento_colaborador,huella_colaborador,estado "+
        "FROM Colaboradores "+
        "WHERE estado='Activo'";

    Colaboradores obj;
    List<Colaboradores> lista =new ArrayList<Colaboradores>();
    Connection cn=ObtenerConexion();
    Statement stmt;
    ResultSet rs;
    try
    {

        stmt = cn.createStatement();
        rs = stmt.executeQuery(sql);
        while ( rs.next() )
        {
            obj=new Colaboradores();
            obj.setCod_Colaborador(rs.getInt("Cod_Colaborador"));
            obj.setCi_Colaborador(rs.getString("Ci_Colaborador"));
            obj.setApe1_Colaborador(rs.getString("Ape1_Colaborador"));
            obj.setApe2_Colaborador(rs.getString("Ape2_Colaborador"));
            obj.setNom1_Colaborador(rs.getString("Nom1_Colaborador"));
            obj.setNom2_Colaborador(rs.getString("Nom2_Colaborador"));
            obj.setDir_Colaborador(rs.getString("dir_colaborador"));
            obj.setTel_Colaborador(rs.getString("tel_colaborador"));

            obj.setFec_Nacimiento__Colaborador(rs.getString("fec_nacimiento_colaborador"));
            //obj.setHuella_Colaborador(rs.getBinaryStream("huella"));
            obj.setEstado(rs.getString("estado"));
            lista.add(obj);

        }

    }

    catch (SQLException e)
    {
        // TODO Auto-generated catch block
        System.out.print(e.getMessage());
    }

    finally
    {
        cn.close();
    }

    return lista;
}

```

## Método de inserción:

```
public String Insertar(Colaboradores colaborador) throws Exception
{
    String result="";
    String sql="INSERT INTO colaboradores(ci_colaborador, "
        + "ape1_colaborador, "
        + "ape2_colaborador, "
        + "nom1_colaborador, "
        + "nom2_colaborador, "
        + "dir_colaborador, "
        + "tel_colaborador, "
        +
        "fec_nacimiento_colaborador, "
        + "estado)"
        + " VALUES
        ("'+colaborador.getCi_Colaborador()+",'',''
        +
        colaborador.getApe1_Colaborador()+",'',''
        +
        colaborador.getApe2_Colaborador()+",'',''
        +
        colaborador.getNom1_Colaborador()+",'',''
        +colaborador.getNom2_Colaborador()+",'',''
        +colaborador.getDir_Colaborador()+",'',''
        +colaborador.getTel_Colaborador()+",'',''
        +colaborador.getFec_Nacimiento__Colaborador()+",'',''
        +colaborador.getEstado()+''');

    Connection cn=ObtenerConexion();
    Statement stmt;
    try
    {
        stmt = cn.createStatement();
        stmt.execute(sql);
        result="Colaborador creado exitosamente";
    }
    catch (SQLException e)
    {
        // TODO Auto-generated catch block
        result="Error en el servidor en la insercion de Colaboradores,
mensaje: "+e.getMessage()+"\n, Causa:"+e.getCause();
        System.out.println(result);
        e.printStackTrace();
    }
    finally
    {
        cn.close();
    }

    return result;
}
```

## Método de actualización:

```
public String Update(Colaboradores colaborador) throws Exception
{
    String result="";
    String sql="UPDATE colaboradores SET
ci_colaborador='"+colaborador.getCi_Colaborador()+"',"
+
"ape1_colaborador='"+colaborador.getApe1_Colaborador()+"',"
+
"ape2_colaborador='"+colaborador.getApe2_Colaborador()+"',"
+
"nom1_colaborador='"+colaborador.getNom1_Colaborador()+"',"
+
"nom2_colaborador='"+colaborador.getNom2_Colaborador()+"',"
+
"dir_colaborador='"+colaborador.getDir_Colaborador()+"',"
+
"tel_colaborador='"+colaborador.getTel_Colaborador()+"',"
+
"fec_nacimiento_colaborador='"+colaborador.getFec_Nacimiento__Colaborador()+"',"
+
"estado='"+colaborador.getEstado()+"' WHERE
cod_colaborador='"+colaborador.getCod_Colaborador();

        Connection cn=ObtenerConexion();
        Statement stmt;
        try
        {
            stmt = cn.createStatement();
            stmt.execute(sql);
            result="Colaborador actualizado exitosamente";
        }
        catch (SQLException e)
        {
            // TODO Auto-generated catch block
            result="Error en el servidor en la actualizacion de Colaboradores,
mensaje: "+e.getMessage()+"\n, Causa:"+e.getCause();
            System.out.println(result);
            e.printStackTrace();
        }
        finally
        {
            cn.close();
        }

        return result;
    }
}
```

## Método de eliminación:

```
public String Delete(int codigo) throws Exception
```

```

{
    String result="";
    String sql="UPDATE colaboradores SET estado='Inactivo' WHERE
cod_colaborador="+codigo;

    Connection cn=ObtenerConexion();
    Statement stmt;
    try
    {
        stmt = cn.createStatement();
        stmt.execute(sql);
        result="Colaborador eliminado exitosamente";
    }
    catch (SQLException e)
    {
        // TODO Auto-generated catch block
        result="Error en el servidor en la eliminacion virtual de
Colaboradores, mensaje: "+e.getMessage()+"\n, Causa:"+e.getCause();
        System.out.println(result);
        e.printStackTrace();
    }
    finally
    {
        cn.close();
    }

    return result;
}

```

#### 4.4. Pruebas de funcionamiento

##### Descripción de las principales interfaces de la aplicación web

**Login:** Permite el ingreso al sistema validando el tipo de usuario para el acceso a las funcionalidades de la aplicación respectivos dependiendo del usuario.

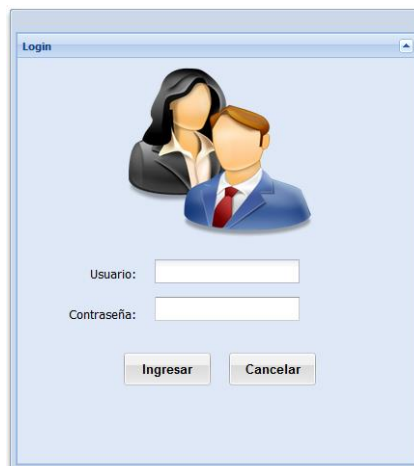




Imagen 21. Login de la aplicación web. Elaborado por el autor

**Página principal:** Permite acceder a la página principal donde se encuentra las funcionalidades de la aplicación web.

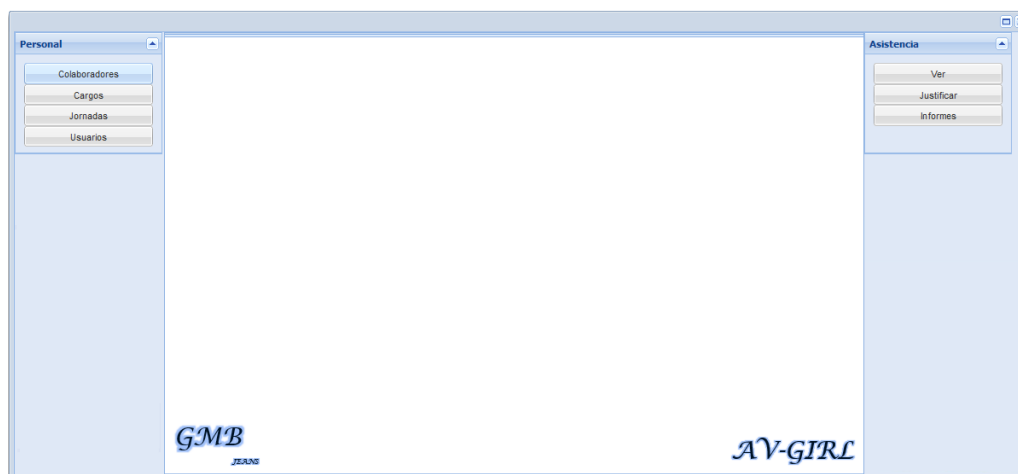


Imagen 22. Página principal de la aplicación web. Elaborado por el investigador

**C.R.U.D.:** Permite ingresar, editar y eliminar información registrada en la base de datos. En la siguiente imagen se muestra la interfaz que será utilizada para registrar la información de los colaboradores incluyendo el registro de la huella dactilar, se puede acceder a esta interfaz a través del botón “Colaboradores” ubicado en el menú desplegable en la esquina superior izquierda.

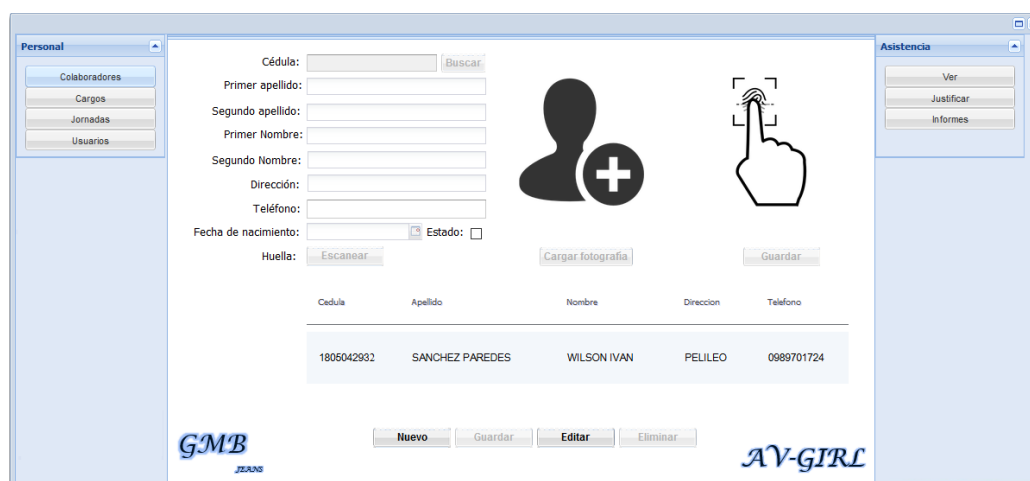
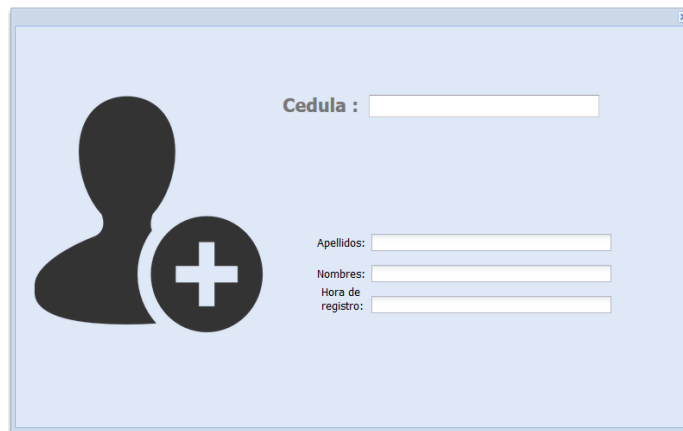


Imagen 23. Página de registro de colaboradores. Elaborado por el investigador

**Registro:** Permite registrar en la base de datos la hora de llegada del colaborador, el mismo debe ingresar su cedula y posteriormente colocar su dedo en el lector de huellas para validar la cedula ingresada, una vez que es validada la huella dactilar se almacena la hora de registro en la base de datos y muestra la información en la interfaz de usuario correspondiente al colaborador que se encuentra registrándose.



Cedula :

Apellidos:

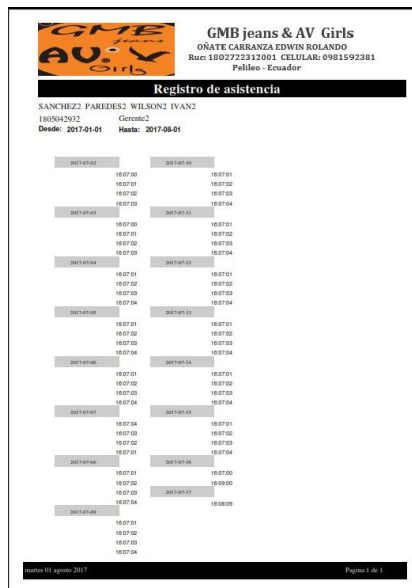
Nombres:

Hora de registro:

Páblon - Ecuador

Imagen 24. Página de registro de asistencia. Elaborado por el investigador

**Reporte:** Formato en que se genera el reporte de la asistencia mensual de un colaborador.



**GMB jeans & AV Girls**  
OÑATE CARRANZA EDWIN ROLANDO  
Ruc: 1102722312001 CELULAR: 0981592381  
Páblon - Ecuador

**Registro de asistencia**

SANCHEZ2 PAREDES2 WILSON2 IVAN2  
180942022 Cúcuta2  
Desde: 2017-01-01 Hasta: 2017-08-01

Desde	Hasta
2017-01-02	2017-01-06
18:07:00	18:07:01
18:07:01	18:07:02
18:07:02	18:07:03
18:07:03	18:07:04
2017-01-09	2017-01-13
18:07:00	18:07:01
18:07:01	18:07:02
18:07:02	18:07:03
18:07:03	18:07:04
2017-01-16	2017-01-20
18:07:01	18:07:01
18:07:02	18:07:02
18:07:03	18:07:03
18:07:04	18:07:04
2017-01-23	2017-01-27
18:07:01	18:07:01
18:07:02	18:07:02
18:07:03	18:07:03
18:07:04	18:07:04
2017-01-30	2017-02-03
18:07:01	18:07:01
18:07:02	18:07:02
18:07:03	18:07:03
18:07:04	18:07:04
2017-02-06	2017-02-10
18:07:01	18:07:01
18:07:02	18:07:02
18:07:03	18:07:03
18:07:04	18:07:04
2017-02-13	2017-02-17
18:07:01	18:07:01
18:07:02	18:07:02
18:07:03	18:07:03
18:07:04	18:07:04
2017-02-20	2017-02-24
18:07:01	18:07:01
18:07:02	18:07:02
18:07:03	18:07:03
18:07:04	18:07:04
2017-02-27	2017-03-01
18:07:01	18:07:01
18:07:02	18:07:02
18:07:03	18:07:03
18:07:04	18:07:04
2017-03-04	2017-03-08
18:07:01	18:07:01
18:07:02	18:07:02
18:07:03	18:07:03
18:07:04	18:07:04
2017-03-11	2017-03-15
18:07:01	18:07:01
18:07:02	18:07:02
18:07:03	18:07:03
18:07:04	18:07:04
2017-03-18	2017-03-22
18:07:01	18:07:01
18:07:02	18:07:02
18:07:03	18:07:03
18:07:04	18:07:04
2017-03-25	2017-03-29
18:07:01	18:07:01
18:07:02	18:07:02
18:07:03	18:07:03
18:07:04	18:07:04
2017-03-31	18:08:00
18:07:01	
18:07:02	
18:07:03	
18:07:04	

Formato 01 agosto 2017 Página: 1 de 1

#### 4.4.1 Pruebas de caja negra

Las pruebas de caja negra se llevan a cabo en la interfaz del sistema web, basándose en las entradas y salidas de información sin tomar en cuenta la ejecución y llamadas internas de la aplicación web.

Estas pruebas tienen como finalidad verificar el comportamiento adecuado de las funcionalidades de la aplicación para que se ajusten a los requerimientos iniciales.

#### Funcionalidad del sistema

<b>Objetivo</b>	<b>Verificar todas las funcionalidades de la aplicación web</b>
<b>Técnica</b>	Verificar los datos de ingreso y salida Generar los resultados esperados en los diferentes procesos
<b>Criterio de complejidad</b>	Las pruebas planteadas se realizaron con éxito y las falencias encontradas fueron corregidas
<b>Consideraciones especiales</b>	Especificar los criterios que puedan influenciar de forma correcta en la ejecución de las pruebas de la aplicación web.

Tabla 23. Pruebas funcionales de la aplicación. Elaborado por el investigador.

## Pruebas de negocio

<b>Objetivo</b>	<b>Asegurar el correcto funcionamiento de la aplicación web basado en el modelo de negocios y la frecuencia de uso</b>
<b>Descripción</b>	<p>Seleccionar el proceso con mayor concurrencia de ejecución y simular las posibles actividades que se van a ejecutar en un lapso de tiempo</p> <p>Se identificará las transacciones que se pueden realizar en un lapso de tiempo.</p>
<b>Técnica</b>	<p>Ejecutar cada caso de uso de manera secuencial como se lo realiza en la vida real.</p> <p>Incrementar el número de transacciones ejecutadas para simular una alta concurrencia en la aplicación WEB.</p> <p>Todas las tareas que indiquen fechas, cantidades serán probados tanto con datos correctos como datos incorrectos.</p> <p>Verificar que los reportes generados muestren la información correcta que se está solicitando.</p> <p>Los mensajes de aviso se muestran en el momento y tiempo adecuado cuando se ejecuta una función.</p>
<b>Criterio de complejidad</b>	Las pruebas planteadas se realizaron con éxito y las falencias encontradas fueron corregidas.
<b>Consideraciones especiales</b>	Las actividades que utilizan cálculos con datos almacenados deben contar con actividades de soporte especial.

*Tabla 24. Pruebas de negocio de la aplicación web. Elaborado por el investigador.*

## Pruebas de Desempeño

<b>Objetivo</b>	<b>Comprobar el tiempo de respuesta del sistema bajo condiciones de alta y baja concurrencia</b>
<b>Técnica</b>	Se comprobó la respuesta del sistema accediendo desde distintos tipos de navegadores.
<b>Criterio de Complejidad</b>	Se estableció un lapso de tiempo dentro del periodo que dura el proceso, en el cual tenga mayor demanda y concurrencia de los colaboradores.
<b>Consideraciones Especiales</b>	Coordinar con los usuarios del sistema, para verificar los posibles errores durante la actividad que lo provoque.

Tabla 25. Pruebas de desempeño de la aplicación web. Elaborado por el investigador.

En la imagen 21 se muestra el resultado de las pruebas.

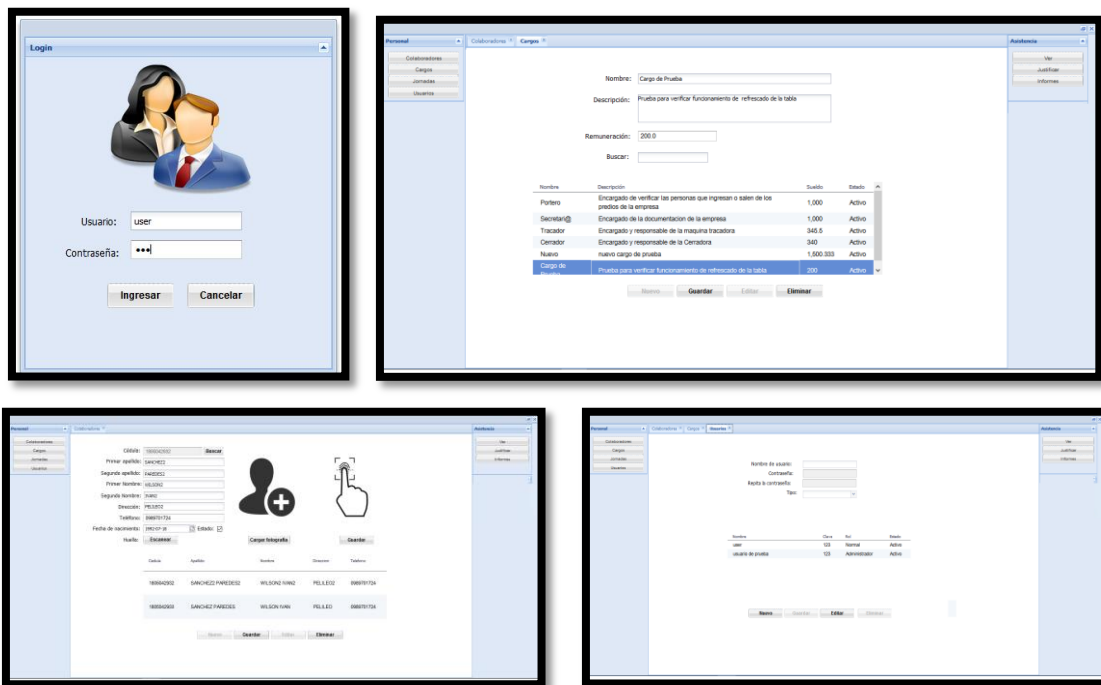


Imagen 26. Funcionamiento de la aplicación web. Elaborado por el investigador.

#### 4.4.2 Pruebas de caja blanca

Se denomina caja blanca al tipo de pruebas que sirven para identificar la funcionalidad interna del código, con la finalidad de comprobar el funcionamiento completo de la aplicación Web.

Para corregir falencias en la estructura interna de la aplicación web se realizan pruebas con los diferentes procesos internos usados con más frecuencia, como son verificación de usuario, ingreso, actualización, eliminación, consultas de datos y reportes

#### Integridad de datos

<b>Objetivo</b>	Asegurar la integridad de los datos ingresados en la aplicación web
<b>Técnica</b>	Registrar y editar información Consultar los datos almacenados para asegurar el correcto almacenamiento.
<b>Criterio de Complejidad</b>	Las funciones del sistema y de la Base de Datos responden según el criterio que fueron diseñadas.
<b>Consideraciones Especiales</b>	Los datos ingresados a usarse deben ser reales Ingresar los datos según la demanda real que requiere el proceso Se debe utilizar un grupo pequeño de información para localizar fácilmente algún evento anormal en la base de datos.

Tabla 26. Pruebas de integridad de datos. Elaborado por el investigador.

En la imagen 22 se muestra los resultados de las pruebas del funcionamiento interno de la aplicación web ejecutando las funciones más usadas como los ingresos, ediciones y eliminación de la información almacenada en la base de datos.

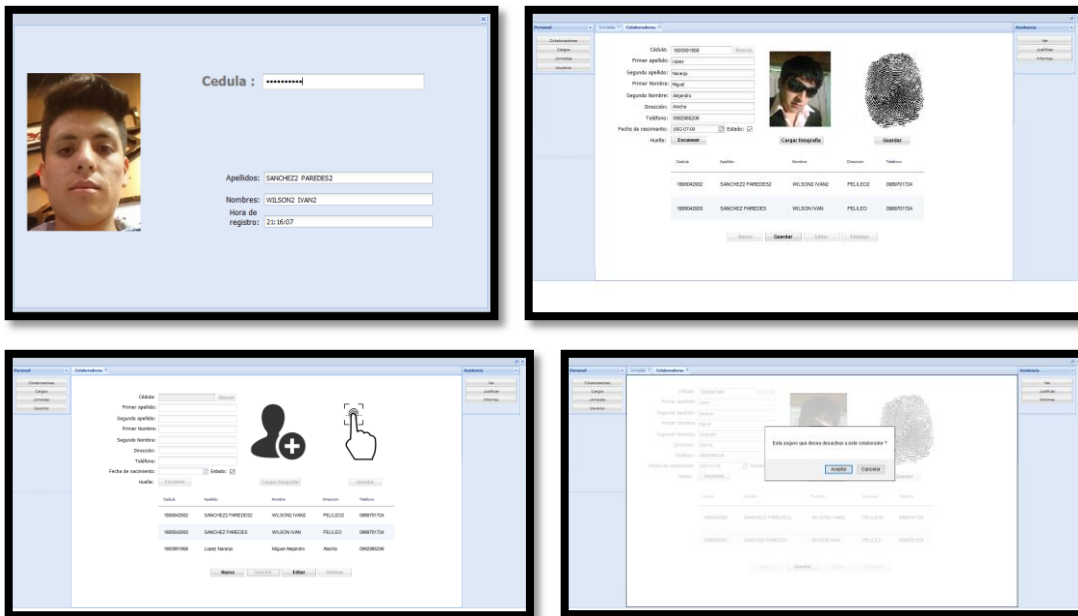


Imagen 27. Pruebas de caja blanca. Elaborado por el investigador.

#### 4.5. Implantación de la Aplicación Web

Actividad	Día 1	Día 2	Día 3	Día 4	Día 5
Organización y distribución de la documentación.					
Preparación del equipo informático y herramientas necesarias para poner en marcha el sistema.					
Puesta en marcha de la Aplicación Web.					
Seguimiento y supervisión del desarrollo de la Implementación.					
Correcciones de las no conformidades mediante acciones correctivas y preventivas.					

Tabla 27. Cronograma de implantación de la aplicación web

#### 4.6. Mantenimiento de la Aplicación Web

El mantenimiento de la aplicación será según lo requiera, ya que a medida del tiempo se irán detectando errores y cambios en el sistema para ser mejorado de acuerdo a los nuevos requerimientos.

#### 4.7. Analisis del proceso de control de asistencia manual vs. control de asistencia automatizado por la aplicación web.

Proceso	Manual	Automatizado
Registro de datos personales del colaborador.	Se registra todos los días en cada ingreso o salida de la jornada laboral.	Se registra una sola vez y es almacenada en la base de datos.
Registro de fecha y hora del momento exacto en que el colaborador ingresa o sale de la jornada laboral.	Se registra observando un reloj y calendario electrónico de pared para verificar la hora y fecha exacta	La fecha y hora exacta es tomada del sistema con una precisión de milisegundos.
Validación de la información	Mediante observación de la secretaria encargada de este proceso, valida y verifica la información	Al registrar una sola vez la información en la base de datos, estos se validan y verifican personalmente para mantener la fiabilidad de la información registrada
Certificación del	Se firma en la zona	Se certifica mediante el



registro de asistencia	designada en el formato impreso.	reconocimiento de la huella dactilar.
Verificación de anomalías en la asistencia	Se observa los registros en el formato impreso y se evalúa si hubo atraso en los ingresos, salidas antes de hora y/o faltas a las jornadas laborales comparando con los horarios establecidos para cada colaborador	Una vez que se registra mediante el lector de huellas internamente se compara los datos del registro con los datos almacenados en la base e internamente evalúa atrasos, salidas antes de hora y/o faltas.
Generación de informes de asistencia mensuales	Se extrae los registros diarios de un archivero y se evalúa uno por uno para obtener un resumen y hacer un informe mensual de cada colaborador de la empresa	Mediante una petición de consulta sql se extrae la información de la base de datos y crea un informe con un formato previamente establecido.

*Tabla 28. Análisis de control de asistencia manual vs. Automatizado. Elaborado por el investigador.*

**4.8. Cuadro comparativo del tiempo necesario antes y después de la implantación de la aplicación web.**

<b>PROCESO</b>	<b>Tiempo</b>	
	<b>Antes de la implantación</b>	<b>Después de la implantación</b>
<b>Registro de datos personales del colaborador.</b>	45 segundos en cada registro de ingreso o salida diaria.	45 segundos una única vez
<b>Registro de fecha y hora del momento exacto en que el colaborador ingresa o sale de la jornada laboral.</b>	10 segundos	0.1 segundos
<b>Validación de la información.</b>	15 segundos	0.1 segundos
<b>Certificación del registro de asistencia.</b>	5 segundos	0.1 segundos
<b>Verificación de anomalías en la asistencia.</b>	20 segundos	0.1 segundos
<b>Generación de informes de asistencia mensuales.</b>	20 minutos por cada colaborador	5 segundos por cada colaborador.

*Tabla 29. Tabla comparativa de tiempos antes y después de la implantación de la aplicación web. Elaborado por el investigador.*

## **CAPITULO V:**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1. Conclusiones**

Finalizado el desarrollo de la aplicación web se puede concluir que:

- El entorno de diseño de las interfaces de usuario GWT y aplicaciones JAVA de escritorio tienen objetos gráficos y configuraciones semejantes, con la diferencia de que ciertas propiedades en GWT pueden ser accedidas únicamente a través de código Java.
- La tecnología GWT se desarrolla por defecto en tres capas: cliente, negocio y servidor, separando las clases y archivos en diferentes paquetes, facilitando el mantenimiento y proporcionando la seguridad necesaria en la transferencia de datos entre capas.
- La aplicación permitió el correcto control de asistencia, evitando y reportando las irregularidades en los ingresos y salidas de las jornadas laborales.
- La librería gráfica “gxt” usada en el desarrollo de las interfaces gráficas de usuario facilitó la configuración y personalización de las propiedades de los componentes usados en las diferentes pantallas de la aplicación.
- El código fuente del controlador del dispositivo se adaptó satisfactoriamente a las necesidades de la aplicación web.

- Se implementó una segunda opción de identificación mediante el ingreso de un código de identidad en caso de requerir reparación o cambio de dispositivo lector de huellas.

## **5.2. Recomendaciones**

- El equipo que hará la función de servidor, debe tener instalado un JRE 1.8 o superior.
- Descargar e instalar los plugins GWT oficiales a través del mismo IDE de desarrollo para cada versión de eclipse y GWT respectivamente y así poder acceder a todos los beneficios del editor gráfico de interfaces.
- Para obtener una interfaz más personalizable se recomienda usar librerías gráficas extras a las nativas de GWT, mismas que se pueden obtener en la página oficial de GWT.
- Se recomienda realizar respaldos periódicos de la base de datos y almacenarlos en medios magnéticos y o en la nube para asegurar la permanencia de la información.

## Bibliografía

- [1] L. R. S. -. L. U. Aragonés, «Google WebToolkit GWT. Compilador Java-toJavaScript.,» *researchgate*, vol. 1, n° 1, p. 5, 2013.
- [2] J. D. Y. C. -. J. A. B. Ricaurte, «Web Application Deveploment Technologies,» *IEEE LATIN AMERICA*, vol. 12, n° 02, p. 6, 2014.
- [3] A. Zeelenberg, «Reconocimiento de huellas dactilares,» [En línea]. Available: <http://www.biometria.gov.ar/metodos-biometricos/dactilar.aspx>. [Último acceso: 22 Agosto 2016].
- [4] E. O. S. -. S. F. Vier, «Programando Redes de Sensores Inalámbricos,» 2011. [En línea]. Available: [http://sedici.unlp.edu.ar/bitstream/handle/10915/19557/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/19557/Documento_completo.pdf?sequence=1). [Último acceso: 21 Agosto 2016].
- [5] C. C. y. J. P. Charalambos, «Método para Adaptar una Librería de,» *LATIN AMERICAN JOURNAL OF COMPUTING*, vol. 2, n° 2, p. 6, 2015.
- [6] H. F. M. V. -. M. S. R. Idrovo, «Repositorio Virtual Spoch,» 2010. [En línea]. Available: <http://dspace.espoch.edu.ec/handle/123456789/552>. [Último acceso: 21 Agosto 2016].
- [7] J. D. Y. C. -. J. A. B. Ricaurte1, «Web Application Deveploment Technologies,» *IEEE LATIN AMERICA TRANSACTIONS*, vol. 12, n° 2, p. 6, 2014.

- [8] J. d. C. F. R. G. M. M. J. V. P. P. López Escobés, «Integración de APIs políglotas de mapas en,» 2013.
- [9] A. J. C. P. --. R. C. Choque, «“PROPUESTA DE INTEGRACIÓN DE GOOGLE WEB TOOLKIT CON GLASSFISH,» *Ciencia de la Ingeniería de sistemas*, vol. 1, n° 1, p. 6, 2014.
- [10] O. M. Y. Borda, «Tecnología Java,» Tecnología Java, 2013. [En línea]. Available: <http://tecnologiajava.blogspot.com/>. [Último acceso: 21 Agosto 2016].
- [11] Ecured, «Ecured.cu,» Ecured, 2012. [En línea]. Available: <http://www.ecured.cu/JDK>. [Último acceso: 21 Agosto 2016].
- [12] A. Silva y I. Ania, *Introducción a la Computación*, 1ra ed., Cengage Learning, 2008.
- [13] J. P. Capdevila., «Las Tecnologías Web para la Gestión del Conocimiento.,» Cuba, 2007.
- [14] M. P. -. D. Villafranca, «inf-cr,» 2008. [En línea]. Available: <http://www.inf-cr.uclm.es/www/mpolo/asig/0708/tutorJavaWeb.pdf>. [Último acceso: 21 Agosto 2016].
- [15] S/N, «Conceptos generales de la arquitectura de aplicaciones web,» *Conceptos generales*, p. 27.
- [16] EcuRed, «EcuRed Conocimiento con todos y para todos,» [En línea]. Available: [http://www.ecured.cu/Arquitectura\\_de\\_tres\\_niveles](http://www.ecured.cu/Arquitectura_de_tres_niveles). [Último acceso: 17 02 2016].

- [17] Ecured, «IDES de programacion,» [En línea]. Available: [http://www.ecured.cu/IDE\\_de\\_Programaci%C3%B3n](http://www.ecured.cu/IDE_de_Programaci%C3%B3n). [Último acceso: 22 Agosto 2016].
- [18] C. Krall, «aprenderaprogramar,» 2012. [En línea]. Available: [http://aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=882:ique-es-y-para-que-sirve-ajax-ventajas-e-inconvenientes-javascript-asincrono-xml-y-json-cu01193e&catid=78:tutorial-basico-programador-web-javascript-desde-&Itemid=206](http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=882:ique-es-y-para-que-sirve-ajax-ventajas-e-inconvenientes-javascript-asincrono-xml-y-json-cu01193e&catid=78:tutorial-basico-programador-web-javascript-desde-&Itemid=206). [Último acceso: 22 Agosto 2016].
- [19] C. Commons, «Creative Commons.net,» [En línea]. Available: <http://es.ccm.net/contents/66-introduccion-bases-de-datos>.
- [20] GFC Aprende Libre, «GFC Aprende Libre,» consultado el 08-12-2015. [En línea]. Available: [http://www.gcfaprendelibre.org/tecnologia/curso/informatica\\_basica/aplicaciones\\_web\\_y\\_todo\\_acerca\\_de\\_la\\_nube/1.do](http://www.gcfaprendelibre.org/tecnologia/curso/informatica_basica/aplicaciones_web_y_todo_acerca_de_la_nube/1.do).
- [21] anmardo1, «RIAs, Aplicaciones Ricas en Internet,» 2010. [En línea]. Available: [http://www.macuarium.com/cms/index.php?option=com\\_content&task=view&id=1950&Itemid=98](http://www.macuarium.com/cms/index.php?option=com_content&task=view&id=1950&Itemid=98). [Último acceso: 22 Agosto 2016].
- [22] M. Fowler, «MartinFowler.com,» 18 Julio 2006. [En línea]. Available: <https://martinfowler.com/eaDev/uiArchs.html>. [Último acceso: 2 Agosto 2017].
- [23] Sparx Systems, «Diagrama de Secuencia UML 2,» consultado el 14-03-2016. [En línea]. Available: [http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_sequencediagram.html](http://www.sparxsystems.com.ar/resources/tutorial/uml2_sequencediagram.html).

- [24] J. S. Perry, «IBM,» IBM, 12 Diciembre 2012. [En línea]. Available: <http://www.ibm.com/developerworks/ssa/java/tutorials/j-introjava1/>. [Último acceso: 21 Agosto 2016].
- [25] H. F. AYALA, «Ventajas y desventajas de Ajax,» 08 Octubre 2007. [En línea]. Available: <http://ventajasydesventajsdeajax.blogspot.com/2007/10/ventajas-y-desventajas-de-ajax.html>. [Último acceso: 22 Agosto 2016].
- [26] S. Ceria, «Casos de Uso,» consultado el 10-03-2016. [En línea]. Available: [http://www-2.dc.uba.ar/materias/isoft1/2001\\_2/apuntes/CasosDeUso.pdf](http://www-2.dc.uba.ar/materias/isoft1/2001_2/apuntes/CasosDeUso.pdf).



## ANEXOS

### ANEXO N° 1

#### Entrevista para el proceso de control y registro de asistencia (Secretaria)

<b>1</b>	<b>¿Qué personas intervienen en el proceso de Registro de la Asistencia?</b>
<b>2</b>	<b>¿Cómo se realiza el proceso de registro?</b>
<b>3</b>	<b>¿Cómo están establecidos los horarios laborales?</b>
<b>4</b>	<b>¿Qué información se obtiene al momento de registrar la asistencia?</b>
<b>5</b>	<b>¿Existen excepciones en el proceso de control de asistencia?</b>
<b>6</b>	<b>¿Qué inconvenientes a tenido al llevar el proceso de registros de asistencia manualmente?</b>

ANEXO N° 2

Formato impreso del registro de asistencia

**GMB FASHION  
& AV GIRLS**  
CONTROL DE ASITENCIA 2017  
JUNIO

OPERARIO: SILVIA MASAQUIZA

DIA	#	HORA ENTRADA	FIRMA	HORA SALIDA	FIRMA	OBSERVACIONES
JUEVES	1	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
VIERNES	2	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
LUNES	5	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
MARTES	6	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
MIÉRCOLES	7	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
JUEVES	8	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
VIERNES	9	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
LUNES	12	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
MARTES	13	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
MIÉRCOLES	14	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
JUEVES	15	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
VIERNES	16	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
LUNES	19	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
MARTES	20	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
MIÉRCOLES	21	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
JUEVES	22	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
VIERNES	23	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
LUNES	26	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
MARTES	27	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
MIÉRCOLES	28	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
JUEVES	29	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
VIERNES	30	8:00	Silvia Masaquiza	17:00	Silvia Masaquiza	
DIAS TRABAJADOS						
TOTAL HORAS TRABAJADAS:						
TOTAL:						

# CONTROL ASISTENCIA

JULIO 2017

DIA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SABADO
	10	11	12	13	14	15
OPERARIO					8:00	
MARLON ANDRADE	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>
SILVIA MASAQUIZA	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	FALTA	FALTA
ALEXANDER ONATE	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>
ESTHELA TIBAN	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>
FERNANDO TIBAN	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>
CARMEN TOANGA	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>
ABIGAIL VARGAS	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>
SERGIO CONDO	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>
SONIA MARTINEZ	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>
TATIANA PERRAZO	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>
JHON PEREZ	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>	8:00 <del>8:00</del>

## ANEXO N° 3

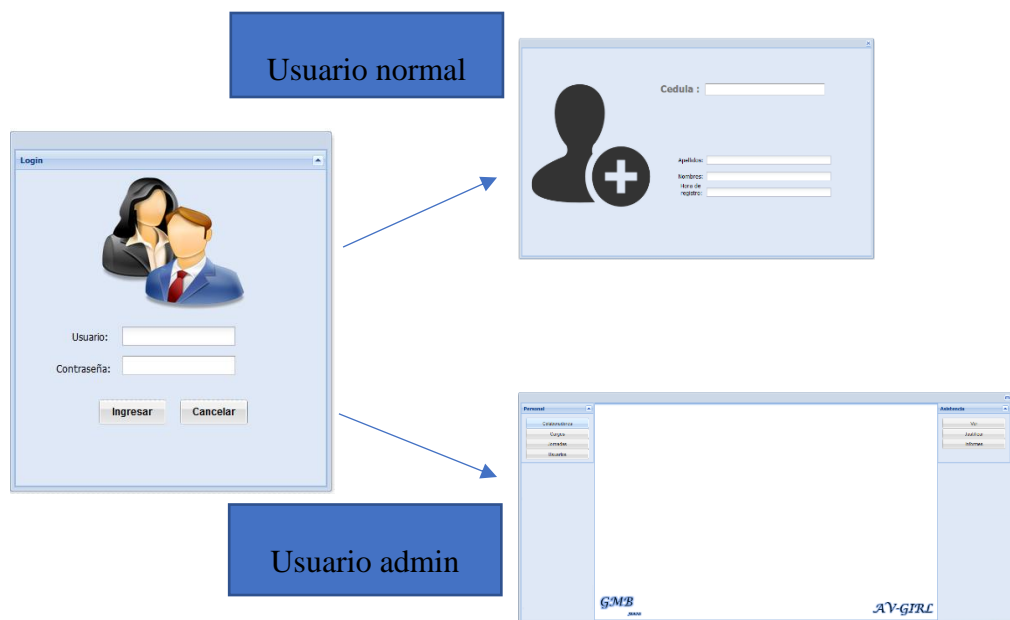
### Manual de usuario

# MANUAL DE USUARIO PARA EL CORRECTO USO DE LA APLICACIÓN WEB DE CONTROL DE ASISTENCIA APLICANDO BIOMETRIA DACTILAR COMO METODO DE IDENTIFICACION

## INICIO DE SESION

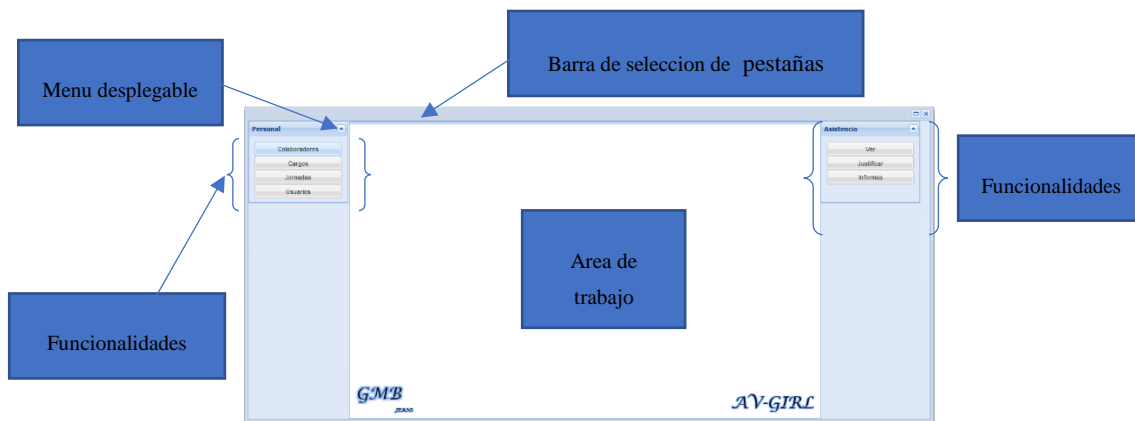
Esta interfaz permite la validación de usuarios habilitados para acceder a las diferentes funcionalidades de la aplicación web, si es un usuario administrador tendrá acceso a las funciones de manipulación de información y generación de informes, en caso de ser un usuario normal tendrá acceso únicamente a la interfaz de registro.

En la siguiente interfaz deberá ingresar un nombre de usuario y una contraseña previamente almacenado en la base de datos.



## PÁGINA PRINCIPAL DE LA APLICACIÓN

A esta página se accede únicamente con un usuario administrador registrado y validado en la base de datos, en donde podemos reconocer el área de trabajo, el menú desplegable, la barra de selección de pestañas, en esta página principal podremos acceder a todas las funcionalidades que nos ofrece la aplicación web.

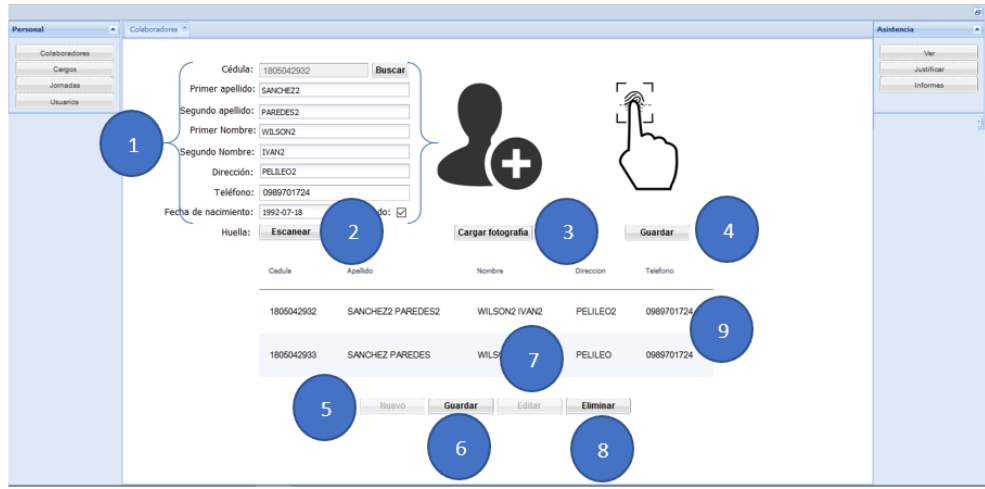


## INTERFAZ DE COLABORADORES

En esta interfaz de usuario podemos ingresar, editar o eliminar los datos personales de un colaborador, incluyendo el registro de su huella dactilar y una fotografía tamaño carnet para tener un registro más detallado del colaborador en cuestión.

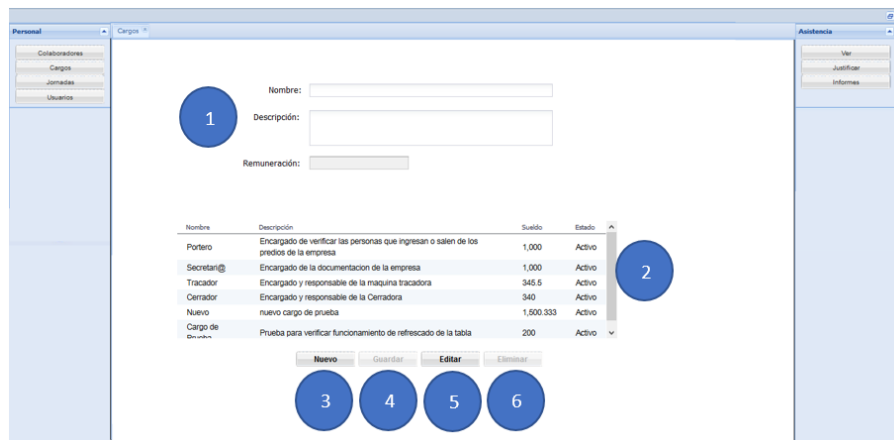
1. **Área de ingreso, edición y visualización de datos personales**
2. **Botón "Escanear"**: Sirve para mostrar la interfaz encargada de escanear validar y almacenar la huella dactilar del colaborador, misma que será usada para verificar la identidad al momento de realizar los registros de asistencia.
3. **Botón "Cargar fotografía"**: Este botón despliega el navegador de archivos para buscar y seleccionar una fotografía en tamaño carnet en formato jpg misma que será almacenada en una ruta de archivo en el disco duro y su url de ubicación será almacenada en la base de datos
4. **Botón "Guardar"**: Este botón almacena la huella dactilar validada
5. **Botón "Nuevo"**: Este botón habilita los componentes gráficos para registrar a un nuevo colaborador.
6. **Botón "Guardar"**: Este botón guarda toda la información del colaborador en la base de datos, previa validación de los mismos.
7. **Botón "Editar"**: Este botón habilita los componentes gráficos y el botón de búsqueda para buscar y editar la información de un colaborador.
8. **Botón "Eliminar"**: Este botón desactiva a un colaborador dentro de la base de datos.

9. **Tabla de colaboradores existentes:** En esta tabla se muestran los colaboradores que se encuentran registrados y activos en la base de datos.



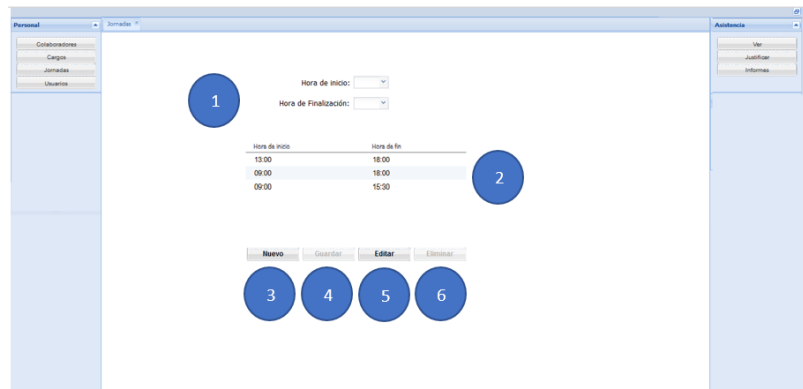
**Interfaz de “Cargos”:** En esta interfaz podemos ingresar, actualizar o eliminar la información de los cargos existentes en la empresa.

1. **Área de ingreso, actualización y visualización de información de los cargos existentes en la base de datos.**
2. **Tabla:** En esta tabla se pueden visualizar los cargos registrados y activos en la base de datos.
3. **Botón “Nuevo”:** Este botón habilita los objetos gráficos para registrar la información de un nuevo cargo a crearse.
4. **Botón “Guardar”:** Este botón guarda los datos registrados en la interfaz en la base de datos
5. **Botón “Editar”:** Este botón habilita los objetos gráficos para editar o visualizar la información de un cargo almacenado en la base de datos.
6. **Botón “Eliminar”:** Este botón desactiva un cargo almacenado en la base de datos.



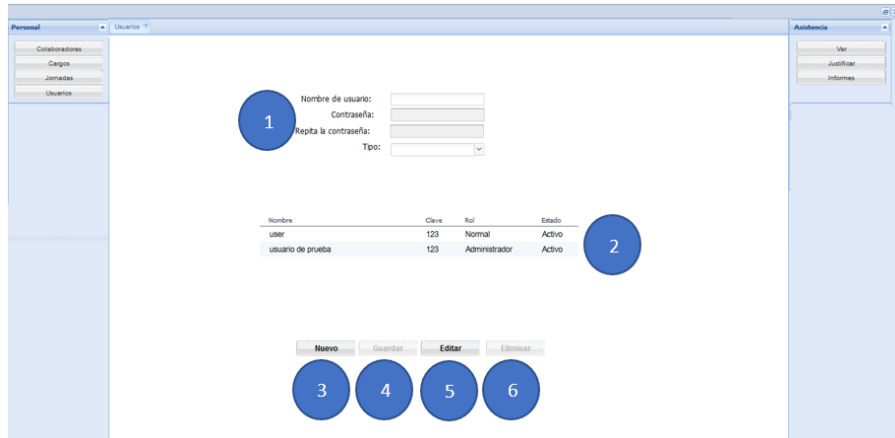
**Interfaz de “Jornadas”:** En esta interfaz podemos ingresar, actualizar o eliminar la información de los cargos existentes en la empresa.

1. **Área de ingreso, actualización y visualización de información de las jornadas existentes en la base de datos.**
2. **Tabla:** En esta tabla se pueden visualizar las jornadas registradas y activas en la base de datos.
3. **Botón “Nuevo”:** Este botón habilita los objetos gráficos para registrar la información de una nueva jornada a crearse.
4. **Botón “Guardar”:** Este botón guarda los datos registrados en la interfaz en la base de datos
5. **Botón “Editar”:** Este botón habilita los objetos gráficos para editar o visualizar la información de una jornada almacenado en la base de datos.
6. **Botón “Eliminar”:** Este botón desactiva una jornada almacenado en la base de datos.



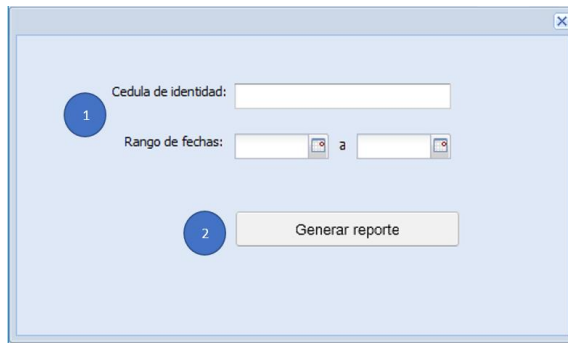
**Interfaz de “Usuarios”:** En esta interfaz podemos ingresar, actualizar o eliminar la información de los usuarios que tendrán acceso a las diferentes funcionalidades de la aplicación web.

1. **Área de ingreso, actualización y visualización de información de los usuarios existentes en la base de datos.**
2. **Tabla:** En esta tabla se pueden visualizar los usuarios registrados y activos en la base de datos.
3. **Botón “Nuevo”:** Este botón habilita los objetos gráficos para registrar la información de un nuevo usuario a crearse.
4. **Botón “Guardar”:** Este botón guarda los datos registrados en la interfaz en la base de datos
5. **Botón “Editar”:** Este botón habilita los objetos gráficos para editar o visualizar la información de un usuario almacenado en la base de datos.
6. **Botón “Eliminar”:** Este botón desactiva un usuario almacenado en la base de datos.



**Interfaz de “Generar reporte”:** En esta interfaz vamos a generar un reporte de los registros de asistencia en base a la cedula de identificación del colaborador y un rango de fechas de donde se extraerá la información de la base de datos.

1. **Área de ingreso de los parámetros de entrada para poder generar el reporte.**
2. **Botón “Generar reporte”:** Este botón valida y envía los parámetros de entrada a la base de datos para extraer mediante una consulta la información necesaria para generar el reporte de asistencia de cada colaborador.



**Registro de asistencia:** En esta interfaz se ingresa la cedula como credencial de identificación para posteriormente escanear la huella dactilar para identificarse y almacenar el registro en la base de datos.

