



Pontificia Universidad  
Católica del Ecuador

## **ESCUELA DE INGENIERIA EN SISTEMAS**

**Tema:**

“Desarrollo de un sistema web de nómina y talento humano para la  
Empresa PRODEGEL S.A.”

Proyecto de investigación y desarrollo previo la obtención del título de  
Ingeniero en Sistemas y Computación

**Línea de Investigación:**

Sistemas de Información y/o Nuevas Tecnologías de la Información y  
Comunicación y sus aplicaciones

**Autor:**

Jorge Esteban González Palacios

**Directora:**

Ing. Mg. Verónica Maribel Pailiacho Mena

**Ambato - Ecuador**

**Abril 2019**

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR  
SEDE AMBATO**

**HOJA DE APROBACIÓN**

**Tema:**

"Desarrollo de un sistema web de nómina y talento humano para la Empresa  
PRODEGEL S.A."

**Línea de Investigación:**

Sistemas de Información y/o Nuevas Tecnologías de la Información y  
Comunicación y sus aplicaciones

**Autor:**

**JORGE ESTEBAN GONZALEZ PALACIOS**



Verónica Maribel Pailiacho Mena, Ing. Mg.  
CALIFICADORA

f.....

Medina Chicaiza Ricardo Patricio, Ing. Mg.  
CALIFICADOR

f.....

Balseca Manzano José Marcelo, Ing. Mg.  
CALIFICADOR

f.....

Mónica Patricia Mena Moreno, Ing. Mg.  
DIRECTORA ESCUELA DE INGENIERÍA EN SISTEMAS

f.....

Hugo Rogelio Altamirano Villarroel, Dr.  
SECRETARIO GENERAL DE LA PUCESA

f.....

  
Pontificia Universidad Católica del Ecuador  
SECRETARÍA GENERAL  
PROCURADURÍA

**Ambato - Ecuador**

**Abril 2019**

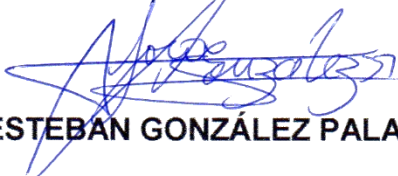
## DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Jorge Esteban González Palacios portador de la cédula de ciudadanía No. 1803071438 autor del trabajo de graduación titulado “Desarrollo de un sistema de nómina y talento humano para la empresa PRODEGEL S.A”, previo a la obtención del título profesional de Ingeniero en Sistemas y Computación, en la escuela de Ingeniería de Sistemas y Computación.

1.- 1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE Ambato, el referido trabajo de graduación, respetando las políticas de propiedad intelectual de Universidad.

Ambato, abril 2019



**JORGE ESTEBAN GONZÁLEZ PALACIOS**

**CC. 1803071438**

## **AGRADECIMIENTO**

A mi esposa, que me ha acompañado durante todo el desarrollo del presente proyecto.

## **DEDICATORIA**

A mis padres, y a mi padrino Fabricio, quienes han sido un apoyo incondicional durante toda mi trayectoria de estudios.

## RESUMEN

Este proyecto documenta el proceso teórico y práctico del desarrollo de la aplicación PRODEGELRRHH, creada para el Área de Talento Humano de la empresa productora de Gelatina Ecuatoriana, con la finalidad de reemplazar el programa de nómina existente que no es actualizable en función de las necesidades del negocio. Los procesos que apoya PRODEGELRRHH son: Gestión del Personal, Gestión de Nómina, y Asistencia, los mismos que fueron seleccionados después de aplicar la Matriz de Vester para identificar los principales problemas del área. Para la obtención de las características de cada uno de los módulos, se aplicó la metodología de entrevistas con el área de talento humano, estas características fueron plasmadas en las listas de características de las iteraciones de desarrollo del sistema. Para el desarrollo de *software*, se aplicó la Metodología *Feature Driven Development*, debido a que se enfoca en la calidad, entrega constante de resultados y monitoreo permanente del proyecto, hechos que otorgan confianza al cliente. Para otorgar tranquilidad al usuario en el uso de la herramienta, se optó por una interfaz *web*, amigable e intuitiva, de muy fácil uso, que visualmente no genera dificultad al operar, pues es similar a las diversas aplicaciones con las que hoy en día se interactúa en gran variedad de dispositivos; y para evitar inversiones altas en licencias se utilizaron herramientas *open source*. El resultado final fue una aplicación que ayudó a manejar los procesos internos del departamento de recursos humanos, reduciendo el tiempo que se empleaba en ellos. Palabras clave: *software*, recursos humanos, nómina, FDD.

## **ABSTRACT**

This project documents the theoretical and practical process for the development of the PRODEGELRRHH application which was created for the human resources department of the Ecuadorian gelatin producer, PRODEGEL S.A. in order to replace the current payroll program that is not updatable according to the needs of the business. PRODEGELRRHH will support the process of staff management, payroll management and attendance, which were selected after applying the Vester matrix to identify the main problems of the department. To obtain the features of each of the modules, the methodology of interviews were applied in the human resources department. These features were reflected in the list of features of the system's development iterations. For the software development, the methodology of Feature Driven Development will be applied since it focuses on quality, constant delivery of results and permanent monitoring of the project, facts that give confidence to the client. To make the tool more user-friendly, an intuitive and easy-to-use web interface was chosen. It does not generate any difficulty when the application is used, since it is very similar to the different applications which are used daily on a wide variety of devices. Open source tools are used to avoid high investments in licenses. The final result was an application that helped to manage all the internal process of the human resources department, thus reducing the time spent on them.

## TABLA DE CONTENIDOS

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD .....	iii
AGRADECIMIENTO .....	iv
DEDICATORIA.....	v
RESUMEN.....	vi
ABSTRACT .....	vii
CAPITULO I.....	1
FUNDAMENTACIÓN TEÓRICA .....	1
1.1. Antecedentes .....	1
1.2. Significado del Problema.....	2
1.3. Definición del Problema .....	3
1.4. Delimitación del Problema.....	5
1.5. Objetivos .....	6
1.6. Justificación .....	7
1.7. Fundamentos Teóricos.....	9
CAPITULO II.....	38
METODOLOGÍA.....	38
2.1. Metodología de Desarrollo .....	38
CAPITULO III.....	55

RESULTADOS .....	55
3.1. FASE 1: Desarrollar un modelo global .....	55
3.1.1. Contexto.....	55
3.1.2. Walkthrough general .....	59
3.1.3. Walkthrough por módulos .....	61
3.1.4. Diagrama de clases .....	69
3.2. FASE 2: Elaborar una lista de características .....	71
3.3. FASE 3: Plan por características.....	74
3.4. FASE 4: Diseño por características.....	78
3.5. FASE 5: Desarrollo por características.....	81
CAPITULO IV .....	82
ANÁLISIS Y VALIDACIÓN DE RESULTADOS.....	82
4.1. Manual .....	82
4.2. Pruebas Integrales .....	113
CAPITULO V .....	119
CONCLUSIONES Y RECOMENDACIONES .....	119
Bibliografía.....	121
GLOSARIO.....	124
ANEXOS .....	125
Anexo 1: Documentos Adicionales – Walkthrough General .....	125

Anexo 2: Documentos Adicionales – Walkthrough por Módulos .....	136
Anexo 3: Desarrollo por características .....	149

## TABLA DE GRÁFICOS

### Figuras

Figura: 1.1. Procesos típicos del Departamento de Talento Humano .....	33
Figura: 1.2. Esquema de Autenticación. ....	34
Figura: 1.3. Equipo Epordo ET90.....	36
Figura: 2.1. Fases de la metodología FDD .....	42
Figura: 2.2. Estructura de una clase .....	46
Figura: 2.3. Diagrama de Secuencia.....	47
Figura: 2.4. Ejemplo de Diagrama de Secuencia .....	48
Figura: 2.5. Secuencia de Desarrollo .....	52
Figura: 3.1. Organigrama General .....	57
Figura: 3.2. Organigrama Talento Humano, Seguridad y Salud Ocupacional .....	58
Figura: 3.3 Árbol de Problemas .....	59
Figura: 3.4. Diagrama de Clases .....	70
Figura: 3.5. Plan de entrega Iteración 1 .....	75
Figura: 3.6. Plan de entrega Iteración 2 .....	75
Figura: 3.7. Plan de entrega Iteraciones 3 y 4.....	76
Figura: 3.8. Plan de entrega Iteración 5 .....	76
Figura: 3.9. Plan de entrega Iteración 6 .....	76
Figura: 3.10. Plan de entrega Iteración 7 .....	77
Figura: 3.11. Plan de entrega Iteraciones 8 y 9.....	77

Figura: 3.12. Plan de entrega Iteración 10 .....	78
Figura: 3.13. Arquitectura Lógica .....	78
Figura: 3.14. Arquitectura Física .....	79
Figura: 3.15. Modelo Lógico de Base de Datos .....	80

## **Tablas**

Tabla: 2.1. Roles Clave .....	40
Tabla: 2.2 Roles de Soporte .....	41
Tabla: 2.3 Roles Adicionales .....	41
Tabla: 2.4 Walkthrough Ejemplo .....	45
Tabla: 2.5. Relaciones de Diagrama de Clases .....	46
Tabla: 2.6. Lista de Características.....	50
Tabla: 3.7. Walkthrough General .....	61
Tabla: 3.8. Walkthrough Empleado.....	65
Tabla: 3.9. Walkthrough Nómina.....	67
Tabla: 3.10. Walkthrough Asistencia.....	68
Tabla: 3.11. Lista de características .....	72

# CAPITULO I

## FUNDAMENTACIÓN TEÓRICA

### 1.1. Antecedentes

En la actualidad, en las empresas, cada vez se hace más necesaria la adquisición de un *software* para la gestión de Talento Humano, que apoye la consecución de los objetivos del área y por tanto los empresariales. Esta tendencia hace que los empresarios busquen opciones en el mercado que se adapten a sus necesidades y presupuesto.

A nivel mundial, sin importar el tamaño de la organización o el número de colaboradores con los que cuenta, se busca *software* de Talento Humano, que se convierta en una herramienta, cuyo principal beneficio, sea dar confianza en la información que se emite, tanto al empleado como al empleador. Otras características, que se evalúan son: automatización de procesos, utilización eficiente de recursos, costo que se adapte al presupuesto, interfaz amigable, cumplimiento de normativas, etc.

En Ecuador, las organizaciones, debido a las constantes regulaciones gubernamentales en relación a la normativa laboral y a la corriente de mejorar

el clima laboral, buscan alternativas de *software* que les permitan gestionar su información y mejorar sus procesos, PRODEGEL S.A. (Productora de Gelatina Ecuatoriana S.A.) es una de ellas, por lo que ha creído conveniente, conseguir un *software*, que le ayude principalmente en el proceso de nómina y adicionalmente en otras actividades que ejecuta el Área de Talento Humano. Debido a que de momento el *software* con el que cuenta, no les está prestando la funcionalidad que requieren y no puede ser actualizado por falta de continuidad del proveedor, por tanto, la gestión de la información se vuelve compleja y en algunos casos inexistente.

Por otra parte para asegurar la permanencia de los colaboradores, es importante convertirse en una organización atractiva, que brinde información clara de ingresos y egresos de salarios, liquidaciones, etc que reciben los empleados, por lo tanto, es fundamental garantizarla a través de la herramienta que se desarrollará.

## **1.2. Significado del Problema**

PRODEGEL S.A. es una industria especializada en la elaboración de gelatina, que con este proyecto se beneficia al recibir la herramienta adecuada, para ejecutar de la mejor forma, los procesos del Área de Talento Humano, de acuerdo con sus necesidades.

La interfaz *web* es una alternativa, amigable, familiar e intuitiva para los usuarios, por tanto, la curva de aprendizaje es baja y corta, lo que es beneficioso para el implementador en inversión de tiempo. El desarrollo de este tipo de aplicaciones, tiene varias ventajas, por ejemplo, la metodología de desarrollo es de ágil aplicación, es multiplataforma, versátil y libre.

Para el desarrollo de la aplicación, se utilizará la metodología FDD (*Feature Driven Development* / Desarrollo Basado en Funciones), debido a las múltiples ventajas que ofrece, de las que se destaca, el hecho de que se basa en la calidad, los entregables son continuos y el monitoreo del proyecto es constante.

### **1.3. Definición del Problema**

#### **1.3.1. Planteamiento del Problema**

No se cuenta con una herramienta que apoye la gestión del Área de Talento Humano, además el *software* que actualmente se utiliza no cubre las necesidades de la Empresa referentes a: actualización y creación de rubros, aplicación ágil de la normativa gubernamental, roles de pago detallados, control de permisos, atrasos e inasistencias y reportes para toma de decisiones oportunas.

### **1.3.2. Preguntas básicas**

#### **¿Quién origina el problema?**

El problema lo origina el personal del Área de Talento Humano, al no contar con el *software* actualizado y necesario para ejecutar las actividades adecuadamente, debido a que el programa con el que cuentan no es actualizable porque la empresa proveedora ya no existe.

#### **¿Por qué se origina el problema?**

Por la falta de mantenimiento al *software* existente, por tanto es caduco para las actividades requeridas y no permite creación de rubros adicionales y los mismos no se calculan de forma automática, es por eso que los valores de los rubros deben calcularse en Excel y cargarse al sistema.

#### **¿Qué lo origina?**

La necesidad de aplicar los continuos cambios de la normativa gubernamental.

#### **¿Cuándo se origina?**

El problema se origina a inicios del año 2016.

## **1.4. Delimitación del Problema**

### **Delimitación Espacial**

El desarrollo del *software* se realizará en la empresa PRODEGEL S.A., para el Área de Talento Humano, enfocado en cubrir el proceso de nómina y actividades adicionales relacionadas con la gestión del recurso humano.

### **Delimitación Temporal**

La presente investigación se efectuará entre los meses de abril de 2017 a enero de 2018.

### **Delimitación Funcional**

Este proyecto va encaminado a desarrollar un sistema *web*, que sirva de herramienta para la gestión de Talento Humano en la Empresa PRODEGEL S.A., cubriendo las necesidades del Área. Además, servirá para aplicar los conocimientos adquiridos en ingeniería de *software*, base de datos y programación.

Para poder realizar el desarrollo del *software*, se utilizará PHP para el aplicativo y PostgreSQL para la base de datos, para elaborar el sistema, se

deberán realizar algunas actividades previas, estas son: investigar, levantar el requerimiento y comprender las necesidades del Área de Talento Humano de acuerdo con los procesos que ejecutan.

Luego se aplicará la metodología FDD para Desarrollo de *Software*, a través de los siguientes pasos, Desarrollar un modelo global, Construir una lista de los rasgos, Planear por rasgo, Diseñar por rasgo y Construir por Rasgo.

## **1.5. Objetivos**

### **1.5.1. Objetivo General**

Desarrollar un sistema *web* para la administración de nómina y talento humano para la empresa PRODEGEL S.A.

### **1.5.2. Objetivos Específicos**

1. Fundamentar teórica y metodológicamente el desarrollo de una aplicación para nómina y talento humano.
2. Analizar los requerimientos específicos y los procesos actuales que realiza el departamento de talento humano de PRODEGEL S.A.
3. Proporcionar reportes con la información requerida al departamento de Talento Humano de PRODEGEL S.A.

## 1.6. Justificación

El interés por resolver con el presente desarrollo, es principalmente conseguir Gestionar el trabajo del Área de Talento Humano, haciendo que la ejecución de los procesos sea eficiente con la ayuda de una herramienta creada a su medida, permitiendo que puedan cumplir con sus objetivos y con la normativa legal.

En la actualidad, el recurso humano es una prioridad para toda empresa, debido a que el gasto producto de volver a la persona ideal para el cargo, es alto, tanto en costo como en tiempo, por ello es necesario mantener un clima laboral ideal, que en muchas ocasiones es medido en función, a la claridad de la información que se recibe en el rol de pagos y en la puntualidad de la entrega del salario. Por tanto, un *software* que ayude a que el proceso sea claro, óptimo y oportuno, en gran medida apoya a mejorar el clima laboral y a retener al personal.

Los resultados adicionales que se obtendrán, de la implementación del *software* son: reducción de tiempo de ejecución de tareas, facilidad en aplicación de cambios legales y control de tiempo efectivo del personal.

Dar solución al requerimiento de PRODEGEL S.A., con un sistema *web* es muestra de la versatilidad y diversos beneficios del lenguaje PHP, que a bajo costo, permite cubrir necesidades de todo tamaño de empresa, adaptándose al presupuesto, que hoy en día en toda sociedad es un factor que pesa en el momento de tomar decisiones y adquirir productos.

La novedad de este desarrollo radica en la aplicación de la metodología FDD y en el uso de PHP, la metodología antes mencionada ofrece al desarrollador un proceso a aplicar de rápida ejecución, que permite tanto a la persona que elabora el aplicativo como al usuario ir viendo avances de forma continua. Por otra parte, PHP presenta al usuario final una plataforma *web*, amigable, familiar e intuitiva, debido a que la mayor parte de la población utiliza páginas *web*, por lo que de cierta forma el cambio de *software* tiene un impacto bajo en la ejecución diaria de los usuarios.

La factibilidad de la ejecución del proyecto, se debe al apoyo del personal del Área de Talento Humano, con sus conocimientos del proceso y de los Directivos de PRODEGEL S.A. que apoyan la implementación del *software*.

## 1.7. Fundamentos Teóricos

### 1.7.1. Software

El *software* según, Pressman (2010), es:

El producto que construyen los programadores profesionales y al que después le dan mantenimiento durante un largo tiempo. Incluye programas que se ejecutan en una computadora de cualquier tamaño y arquitectura, contenido que se presenta a medida que se ejecutan los programas de cómputo e información descriptiva tanto en una copia dura como en formatos virtuales que engloban virtualmente a cualesquiera medios electrónicos (p. 1).

Se puede concluir que el *software* es un conjunto de instrucciones lógicas, que se ejecutan en cualquier equipo de cómputo e incluso en otros dispositivos para brindar ayuda al usuario.

Pressman (2010), también menciona que: “El *software* se ha incrustado profundamente en casi todos los aspectos de nuestras vidas y, como consecuencia, el número de personas que tienen interés en las características y funciones que brinda una aplicación específica ha crecido en forma notable.” (p.10).

De lo publicado, se comprende que, el *software* ha llegado a convertirse en una herramienta importante y necesaria en las empresas, llegando a ser la columna vertebral de las mismas, debido a los datos y procesos que se ejecutan a través de este.

### **1.7.2. Ingeniería del Software**

Martínez, Laínez, Durango y Ramos (2017), la definen como una forma de ingeniería que se basa en los fundamentos de la ciencia de la computación y de la matemática para conseguir soluciones a problemas cotidianos a través de la generación de *software*. Cuando aparecieron, eran aplicaciones muy pequeñas debido a las restricciones de la arquitectura del hardware de ese entonces. El *software* ha ido mejorando a medida que el hardware ha incrementado sus prestaciones

La Ingeniería de *Software*, es una alternativa para gestionar diversos tipos de actividades de la vida diaria en el planeta, a través de esta se pueden generar herramientas tecnológicas que optimizan la utilización de recursos.

### **1.7.3. Metodología de Desarrollo de Software**

Para Leyva, Prieto, Sampalo y Garzón (2006), la metodología de desarrollo es: “una recopilación de técnicas y procedimientos estructurados en fases

para la producción de productos *software* de manera eficaz y englobando todo el ciclo de vida del *software*" (p. 69).

La Metodología de *Software*, por tanto, es una herramienta de utilidad para los desarrolladores, que les permite ir ejecutando el desarrollo a través de diferentes etapas, las mismas que son documentadas y que de forma organizada atraviesan el ciclo de vida del *software*. Existen diversas metodologías, la selección se hace de acuerdo con varios criterios que se evalúan antes de iniciar el proyecto.

#### **1.7.3.1. Metodologías Ágiles**

Según Lainez (2014):

El término de Metodologías Ágiles se hizo popular cuando diecisiete especialistas en desarrollo de *software*, presentaron los métodos *Extreme Programming (XP)*, *Scrum*, *Feature Driven Development FDD*, entre otros, establecieron principios comunes compartidos por todos. El resultado fue la creación de la Alianza Ágil (*Agile Alliance*) y el establecimiento del Manifiesto Ágil o Agile manifiesto, en el año de 2004. Las metodologías ágiles varían en sus prácticas y en sus fases, sin embargo, comparten algunas características, tales como: desarrollo iterativo e incremental, comunicación y reducción de productos intermediarios y de la documentación extensiva (p. 113 – 114).

El contenido citado indica que, existen varias metodologías ágiles para aplicar, que guardan características similares, enfocadas en acelerar el desarrollo de *software*, sin perder de vista la calidad y el cumplimiento de los requerimientos del usuario.

#### **1.7.10.1. Principios del Manifiesto Ágil**

A continuación, se resumen los principios que describe, Jurado (2010):

- Lo más importante es satisfacer al cliente, con entregas tempranas y permanentes de *software* funcional.
- Los requerimientos de cambios y ampliaciones son aceptados, incluso en las etapas finales del desarrollo. Los cambios que se aplican en muchas ocasiones dan ventajas competitivas al cliente.
- Se entrega soluciones de *software* funcionales, como mínimo cada dos semanas y hasta en cuatro semanas, dependiendo de la complejidad.
- Los usuarios líderes y los desarrolladores tendrán comunicación permanente y trabajan juntos durante todo el proyecto.
- El empoderamiento al equipo es permanente a todo nivel.
- Los procesos ágiles apoyan el desarrollo de *software* sostenible.
- Se requiere que el equipo mantenga un ritmo constante de trabajo.
- La simplicidad es fundamental.

Los principios que se promulgan en la utilización de las metodologías ágiles, tienen énfasis en brindar un buen servicio al cliente, haciendo entregas permanentes que le den confianza en que el proveedor está trabajando continuamente en brindarle soluciones a tiempo, además empoderan a los desarrolladores de tal forma que se promueve una mejora continua en el equipo de trabajo.

#### **1.7.4. Base de Datos**

En su publicación Date (2001), define a la base de datos, como:

Un sistema de base de datos es básicamente un sistema computarizado para llevar registros. Es posible considerar a la propia base de datos como una especie de armario electrónico para archivar, es decir un depósito o contenedor de una colección de archivos de datos computarizados (p. 29).

Coronel, Morris y Rob (2011), menciona lo siguiente al respecto:

Por lo general, una eficiente administración de datos requiere el uso de una base de datos de computadora. Una base de datos es una estructura computarizada compartida e integrada que guarda un conjunto de:

- Datos del usuario final, es decir, datos sin elaborar que son de interés para el usuario final.

- Metadatos, o datos acerca de datos, por medio de los cuales los datos del usuario final son integrados y manejados (p. 7).

Es así, que se comprende que una base de datos es un repositorio de información, clasificada y organizada, la misma es útil para los usuarios finales.

#### **1.7.4.1. PostgreSQL**

En la publicación de Zea, Molina y Redrován (2017), se menciona que: “es un sistema de gestión de base de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado” (p. 12).

Es comprensible que PostgreSQL, sea utilizado frecuentemente, pues no tiene restricciones de uso y es un motor de base de datos muy potente, con excelentes prestaciones.

Para Martín (2011), usar bases de datos es:

Parte fundamental de la gestión de la información empresarial. PostgreSQL se perfila como una alternativa de *software* libre para estas exigencias. Postgres surgió como continuación del proyecto gres para el desarrollo de un sistema de bases de datos, que ha continuado bajo la filosofía de desarrollo de *software* libre. Ha sido descrito como un

sistema de bases de datos relacional orientado a objetos (ORDBMS por sus siglas en inglés) basado en POSTGRES 4.2. Desarrollado en el departamento de ciencias de la computación de la Universidad de Berkeley en California (p. 4).

La filosofía de *software* libre ofrece alternativas con características y prestaciones excelentes, para los desarrolladores, convirtiéndose en elementos poderosos y de gran ayuda, como el caso de PostgreSQL para la gestión de base de datos.

#### **1.7.4.1.1. Ventajas de usar PostgreSQL**

Del sitio oficial [www.postgresql.org](http://www.postgresql.org) (s.f.), se resume lo siguiente

- Ningún usuario puede ser demandado por incumplimiento de acuerdo de licencia, ya que este motor de base de datos no tiene ningún costo relacionado con su utilización.
- Cuenta con una comunidad de profesionales, de quienes se puede aprovechar su conocimiento en soluciones a problemas.
- Ahorro significativo de costos de personal, al no requerir de mantenimientos y ajustes complicados o excesivos.
- Alta fiabilidad y estabilidad.
- Es multiplataforma.
- Ha sido diseñada para entornos de alto volumen de datos.

Los beneficios que brinda PostgreSQL al usuario, son excelentes sin verse afectados por ser un *software* libre, además cuenta con una comunidad colaborativa de usuarios, bastante efectiva para incrementar el conocimiento y de respuesta ágil a soluciones de problemas y mejora del *software*.

#### **1.7.5. Lenguajes de programación**

De la definición de lenguaje de programación, dada por Rodríguez (2003), se resume que, es un lenguaje artificial que sirve para formular programas de computadora. Para facilitar la actividad de desarrollo, existen a disposición de los programadores, lenguajes de alto nivel que no dependen del diseño definido de cada ordenador, sino que son versátiles. Al elaborar un lenguaje de programación es importante definir:

1. Grupo de símbolos y palabras clave disponibles.
2. Criterios gramaticales para elaborar sentencias que sean sintáctica y semánticamente correctas.

En resumen, se comprende que un lenguaje de programación, es un conjunto de instrucciones que son interpretadas por el ordenador, para realizar funciones determinadas.

### 1.7.5.1. Lenguaje HTML

Para Cobo, Gómez, Pérez y Rocha (2005), “HTML es un lenguaje de descripción de hipertexto compuesto por una serie de comandos, marcas o etiquetas también denominadas “Tags” que permiten definir la estructura lógica de un documento *web* y establecer los atributos del mismo” (p. 57).

En resumen, el lenguaje HTML permite crear documentos *web* y poner atributos, como: color de texto, tamaño de fuente, objetos multimedia, etc.

A continuación, se resumen algunos conceptos básicos, con relación a HTML, definidos por Mora (2002),

**Información por hipertexto:** Son varios elementos (texto o imágenes) que se visualiza en pantalla y que están vinculados con datos que pueden estar en otras fuentes, para llegar a las otras fuentes bastará con hacer clic sobre ellos.

**Gráfico:** En la imagen aparece al mismo tiempo, texto, imágenes e incluso sonidos.

**Global:** Que puede ser accedido, desde cualquier tipo de plataforma y lugar geográfico, utilizando cualquier navegador.

**Pública:** Toda su información está compartida en varios equipos de cómputo, que ofertan un lugar para almacenarla. Todos estos datos son públicos y pueden ser extraídos por los usuarios.

**Dinámica:** La información, permite actualizaciones de los usuarios, a pesar de estar almacenada.

**Independiente:** Debido a la gran cantidad de fuentes, es independiente y libre.

#### **1.7.5.2. Lenguaje PHP**

De la información, encontrada en el libro de Cobo et al. (2005), se comprende que PHP es un lenguaje muy potente, modular, versátil y robusto. Los programas desarrollados con PHP, son embebidos por código HTML y corridos por el servidor *web*, a través de un intérprete, previo a devolver al cliente la acción solicitada. Además, el uso de PHP, ha crecido de manera acelerada, debido a su filosofía open source, la facilidad de conectarse con bases de datos y las numerosas funcionalidades y excelentes prestaciones que ofrece.

El Lenguaje PHP, es de amplia utilización debido a que no requiere licenciamiento, lo que hace que no se requiera de gran presupuesto para desarrollos.

Algunas características de PHP, según Cobo et al. (2005), se resumen a continuación

- Los navegadores no toman en cuenta los cambios de línea incrustados en el código HTML, así como la presencia de más de un espacio en blanco entre palabras.
- Cualquier comando que no tenga coherencia es ignorado por el navegador sin emitir ningún mensaje de error.
- Las distintas versiones de navegadores, generan que en ocasiones cada uno actúe de forma diferente, por lo que la visualización de una misma página puede variar de un navegador a otro si no se utilizan los estándares del lenguaje.
- Los comandos van insertados entre los signos < > y suelen tener una orden de inicio y otra de. Ejemplo, para hacer que el texto se muestre en negrita se aplica <B> para señalar el inicio del texto y el comando </B> para el final.
- Al generar sentencias HTML, no importa si se utilizan mayúsculas o minúsculas, pues funcionan igual sin ningún problema.
- Las etiquetas pueden anidarse y acoplarse, produciendo como resultado la suma de todas las sentencias registradas. Por ejemplo, si se requiere que un texto se muestre en negrita y subrayado se podría registrar un anidamiento de etiquetas:  
  
<B><U> El texto deseado </U></B>
- Las sentencias pueden requerir parámetros, que especifican la forma de visualización. Por ejemplo, el comando <HR> crea una línea horizontal que

sirve para separar partes de una página; esta sentencia puede llevar los parámetros ALIGN, que determina la posición (centrada, izquierda, derecha), SIZE, que establece el grosor de la línea en píxeles, y WIDTH, que establece la longitud de la línea.

- Si las sentencias necesitan parámetros, los mismos pueden establecerse en cualquier orden, el valor del parámetro se establece después del signo igual, cuando se registra más de uno, los parámetros se separan por espacios en blanco, Por ejemplo, las siguientes sentencias realizan el mismo trabajo:

```
<HR SIZE=20 WIDTH=40 ALIGN=center>
```

```
<HR ALIGN=center WIDTH=40 SIZE=20>
```

- Los valores de los parámetros deber ser dados entrecomillados, salvo en el caso de que estos valores no incluyan espacios en blanco o caracteres especiales.

El lenguaje HTML, es de fácil aplicación, ofrece al usuario diversas opciones de diseño, a través de la aplicación de estilos para visualización en el navegador.

### **1.7.5.3. Lenguaje JAVASCRIPT**

De la información de Mora (2002), se resume que este, es un lenguaje interpretado, que se basa en objetos y además es multiplataforma. Apareció

en 1995, su núcleo contiene una cadena de objetos, y un grupo de elementos de lenguaje, que pueden ser ampliados añadiendo nuevos objetos. Las aplicaciones que se crean con este lenguaje, se ejecutan por medio de Internet, basándose en el modelo cliente / servidor.

JAVASCRIPT, está orientado a objetos y es muy dinámico, lo que es útil para diversas aplicaciones, además el poder ejecutarlo en *web* lo hace sencillo de usar para el usuario final.

#### **1.7.5.4. Lenguaje CSS**

La definición, que Mora (2002), da para este lenguaje, dice: “Tecnología empleada en la creación de páginas *web*, que permite un mayor control sobre lenguaje HTML. Permite crear hojas de estilo que definen cada elemento, como por ejemplo la forma en que se deben mostrar los encabezados o los enlaces” (p. 27)

Por tanto, se puede entender que es un lenguaje, que permite dar formato y asignar un estilo al diseño *web*.

De lo publicado por, Schultz (2009), se resume lo siguiente

Las hojas de estilo en cascada (Cascading Stylesheets) brindan características extras que ayudan a engrandecer el lenguaje HTML en la

representación gráfica de las páginas *web*. Este lenguaje, se empezó a utilizar en 1996, para definir el formato de los componentes HTML, por ejemplo, ayuda a enlazar los documentos HTML con “plantillas de documento”, guarda la información acerca de las particularidades de los elementos visuales de la página, ayuda a apartar totalmente la estructura de contenidos de su representación y presentación actuales, en la pantalla de cualquier dispositivo.

### **1.7.6. Plataforma de Desarrollo de Software**

#### **1.7.6.1. Eclipse**

De la información publicada, en el sitio oficial [www.eclipse.org](http://www.eclipse.org) (s.f.), se resume que, Eclipse es una plataforma de *software*, que apareció en noviembre de 2001, fruto de un proyecto en el que intervinieron empresas líderes en la industria del *Software*. Eclipse es un grupo de herramientas de programación de código abierto.

Originalmente fue desarrollada por IBM, y a partir del 2003 se creó la Fundación Eclipse para su gestión, con la finalidad de mantener una comunidad de usuarios para: generación y apoyo en proyectos de *software*, mejorar la plataforma y sus productos complementarios.

### 1.7.7. Framework

La definición de Framework, dada por Granados (2014), dice que es: “un patrón/conjunto de bibliotecas orientado al desarrollo de *software*, aunque también puede servir para otros objetivos. Enfocado a certificación de componentes se aplica más a un conjunto bien definido de métricas y técnicas (p. 110).”

Este entorno de trabajo, es un conjunto de prácticas, definiciones y normas aplicadas en el desarrollo de *software*.

#### 1.7.7.1. Bootstrap

Es un Framework, que según Moro y Rodés (2014), es: “para el desarrollo *web*, basado en CSS3 y JavaScript, desarrollado por el equipo de programadores de Twitter y a disposición pública. Contiene multitud de elementos que facilitan el desarrollo *web* y lo hacen más eficaz” (p. 103).

Este Framework, es una herramienta útil para el desarrollo *web*, que se utiliza ampliamente, debido a que los elementos que la conforman, son de gran ayuda con excelentes prestaciones.

Del sitio oficial, [getbootstrap.com](http://getbootstrap.com) (s. f.) se resume lo siguiente

Bootstrap, es un conjunto de herramientas de código abierto que se utiliza para el diseño *web*, pone a disposición del desarrollador, plantillas de diseño, botones, formularios, menús de navegación, etc. que son de gran ayuda para el ahorro de recursos. Es compatible con la mayoría de los navegadores *web* y dispositivos pues su diseño gráfico, es de fácil acoplamiento.

#### **1.7.7.2. CodeIgniter**

Del sitio oficial [ellislab.com](http://ellislab.com) (s.f.), que es la empresa creadora, se resume que CodeIgniter es una biblioteca, que sirve de instrumento para que los desarrolladores de PHP, de nivel medio en adelante, creen aplicaciones *web*, de forma sencilla y ágil. Utiliza un estilo de diseño, en tres capas: modelo, vista y controlador. Tiene múltiples ventajas, entre las que se destacan: un núcleo ligero que permite el procesamiento rápido de las páginas, reutilización de código y gran cantidad de documentación y espacios colaborativos en la red.

CodeIgniter es una herramienta, que según Eslava (2016), es “Open Source y muy pequeño, con una estructura de sus librerías muy bien estructurado. Destaca por lo liviano que es y por su facilidad para ponerlo en marcha. Solo hay que descomprimirlo y ponerlo en una carpeta” (p. 119).

Este instrumento de desarrollo, brinda gran apoyo, a los programadores *web* pues ofrece una alternativa de código abierto y altamente colaborativo, para desarrollo ágil y de fácil implementación.

#### **1.7.8. Lenguaje de modelado Sistemas de Software**

Flores (2005) acerca de UML dice: “Para el modelado de la solución del problema se utiliza el Lenguaje Unificado de Modelado (*Unified Modeling Language*, UML), el cual es una herramienta usada para describir clases, objetos y sus relaciones” (p. 30)

UML, es un lenguaje que permite dar una imagen clara y específica del modelo a desarrollar, a través de la construcción de clases entrelazadas entre sí.

De la publicación de Fowler y Scott (1999), acerca de UML, se resume lo siguiente

Apareció a finales de la década de 1980, de la agrupación de los métodos de Rumbaugh, Booch y Jacobson. Es importante aclarar que el UML, es un lenguaje de modelado, en este, se apoyan los métodos, para expresar los diseños. El lenguaje de modelado, es la clave para que diferentes usuarios puedan comprender un diseño de *software*, sin necesidad de conocer el proceso de elaboración del mismo.

## **1.7.9. Herramientas de modelado**

### **1.7.9.1. GanttProject**

El sitio oficial [www.ganttproject.biz](http://www.ganttproject.biz) (s.n.), informa acerca de GanttProject, lo que se resume a continuación

Fue ideado en el 2003, por Alexandre Thomas, para su desarrollo y mejoras muchas personas han participado en el proyecto. Es una herramienta gratuita, para graficar Diagramas de Gantt, los mismos que se utilizan para representar la planificación, cumplimiento de actividades y asignación de recursos en períodos determinados de tiempo.

### **1.7.9.2. Draw.io**

Acerca de draw.io, el sitio web de Chrome (s.n.), informa que:

Es una aplicación de diagramación gratis de Google Drive (TM) que le permite dibujar:

- Diagramas de flujo.
- Lenguaje Unificado de Modelado.
- Diagrama Entidad-Relación.
- Diagramas de red.
- Modelos de proceso de negocios.

- Organigramas
- Circuitos electrónicos.
- Wireframes y maquetas.

Características:

- Cliente HTML 5 nativo con soporte full de IE 6-8.
- Gran biblioteca de plantillas incorporada.
- Interfaz intuitiva de arrastrar y soltar.
- Función de buscar y agregar imágenes.
- Exportar a PNG/JPG/XML/SVG/PDF.
- Soporte de dispositivos táctiles.
- Colaboración en tiempo real.
- Insertar diagramas en blogs o wikis.

La herramienta brinda al usuario, varias ventajas con facilidad de uso, debido a que su interfaz es amigable e intuitiva, permite dibujar diversos tipos de diagramas y compartirlos de varias formas.

#### **1.7.10. Gestión del Talento Humano**

En la publicación de Montes y González (2006), se indica que, la gestión de Talento Humano: “tiene como finalidad organizar, desarrollar y poner en funcionamiento a las personas que trabajan en una empresa para conseguir

de forma eficiente y eficaz los objetivos de la organización, utilizando para ello una serie de funciones y actividades” (p. 33).

El trabajo del área de Talento Humano en las empresas, abarca un conjunto de actividades, que buscan a través del recurso humano, garantizar el cumplimiento de los objetivos de la institución, para ello se encarga de su selección, evaluación, capacitación, etc.

#### **1.7.10.1. Las Tecnologías de la Información y su aplicación para la Gestión del Talento Humano**

González (2005), menciona al respecto, que en las empresas:

El objetivo final que se persigue es interiorizar la realidad en la que se encuentran inmersas actualmente las empresas, ya que operan en una economía globalizada donde el único factor competitivo sostenible radica en el capital humano, de manera que deberían tomar conciencia de la importancia de las nuevas tecnologías de la información y comunicación entendidas como un instrumento estratégico fundamental a la hora de gestionar los recursos humanos de las organizaciones (p. 1).

Es comprensible entonces, que las TIC hoy en día, sean una herramienta mandatoria en la mayoría de las empresas, pues con su utilización alineada a

los objetivos de las mismas, hacen que el trabajo diario y el uso de recursos sea el adecuado. La aplicación de las mismas, puede ejecutarse en cualquier área o proceso.

De la publicación de Goñi (2012), se resume lo siguiente

El talento humano, la tecnología y el tiempo, son tres fuerzas que se alinean y complementan, para la gestión empresarial, existen a fin de mantener viva a la empresa y dependen unas de las otras.

En primer lugar, el talento humano, es la necesidad básica de la empresa, con su capacidad creativa y ejecutora, le permite distinguirse de entre las demás en el presente y perfilarse hacia un futuro exitoso, el recurso humano talentoso aprende rápido, resuelve problemas con mayor agilidad de tal forma que puede actuar antes y mejor, sin dejar pasar las oportunidades, aplicando siempre la innovación.

En segundo lugar, está la tecnología, que, al incorporarse a la empresa, genera una revolución, complementando las acciones del talento humano, como materia prima o input en un proceso constante, que genera productos complejos, con calidad y reducción de recursos. La tecnología permite generar y almacenar productos creados por unos, para mejorarlos y ponerlos a servicios de otros en procesos productivos o en usos personales, que permiten entregar al cliente, productos de alto valor, en un circuito de mejora continua que asegura el desarrollo y permanencia en el futuro.

Y en tercer lugar, está el tiempo, que es una variable y punto de conexión, entre el talento humano disponible y la tecnología que la empresa puede adquirir, generar valor por medio del talento y la tecnología, es un reto continuo de la empresa pues debe abrirse paso a través de los depredadores tradicionales del tiempo, como por ejemplo: los reprocesos, los errores continuos y no corregidos, las pérdidas de tiempo en formarse para no aplicar los conocimientos, los tiempos utilizados en inspecciones sin valor, el tiempo desperdiciado en conflictos innecesarios, el tiempo perdido por decisiones erróneas, etc.

El talento humano, la tecnología y el tiempo, funcionan como una máquina, son dependientes entre sí, para producir resultados favorables. El aporte de cada parte debe ser equilibrado, a fin de que se ejecuten los procesos adecuadamente con la inversión necesaria de recursos y el aprovechamiento de los mismos al máximo.

#### **1.7.10.2. Procesos del Departamento de Talento Humano**

A continuación, un resumen de lo que, al respecto, Chiavenato (2002), menciona en su publicación

**Admisión de personas.** La función de este proceso, es la selección del personal idóneo para formar parte de la organización, a través de actividades

definidas para insertar nuevos colaboradores en la empresa. Casi siempre esta selección, es realizada por Psicólogos o Sociólogos.

**Aplicación de personas.** Es el área responsable de los cargos y salarios, define ¿Qué deberán hacer las personas?. Para ello se utilizan técnicas especializadas para diseñar los descriptivos de funciones de los empleados, rutas de carrera, y acompañarlos en su desempeño. Para la ejecución de estas actividades, se recomienda a personas con funciones de Estadísticos, analistas de cargos y salarios, etc.

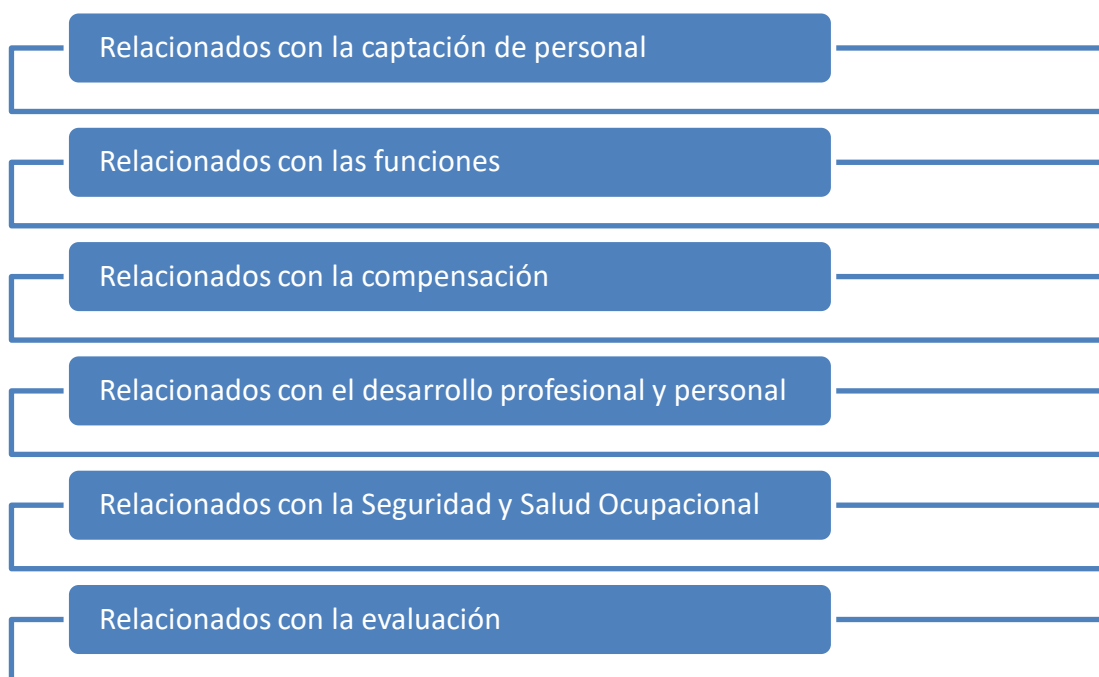
**Compensación de las personas.** Una arista sensible en toda empresa es la referente a los beneficios sociales, en este proceso se indica ¿Cómo compensar a las personas?, se encarga de: salarios, bonificaciones y servicios sociales, por lo general las personas que ejecutan estas funciones son Trabajadores sociales, Analista de nómina, etc.

**Desarrollo de personas.** Esta sección apoya a los colaboradores, de forma que sienten que la organización se preocupa por ellos, y con eso se mantiene un clima laboral adecuado, es una manera de compensar el esfuerzo de los empleados. Ejecuta procesos de capacitación, desarrollo de los colaboradores, programas de ruta de carrera y procesos de comunicación interna y de integración de las personas. Los cargos que se alinean a estas tareas son: Analistas de capacitación, Instructores / Formadores, Comunicadores, etc.

**Mantenimiento de personas.** Normalmente en esta área se desarrollan actividades de calidad de vida, higiene, seguridad y salud ocupacional, las personas que las ejecutan son: Médicos Ocupacionales, Enfermeras, Ingenieros de seguridad y salud ocupacional, etc. Todos ellos encaminan su trabajo, para retener a las personas en el trabajo, a través de procesos que ofrecen condiciones ambientales y psicológicas agradables para la ejecución de las actividades de los empleados.

**Evaluación de personas.** La aplicación de técnicas para evaluar el desempeño de los colaboradores es importante, a fin de verificar el cumplimiento del descriptivo de funciones y controlar sueldos, casi siempre son ejecutadas, Auxiliares de Personal, Consultoría Externa, etc.

Es comprensible, que las organizaciones requieran de Áreas de Talento Humano, que apoyen las estrategias empresariales, debido a que existen diversos procesos, a ejecutarse con relación al personal, y que se agrupan de la siguiente forma:



**Figura: 1.1. Procesos típicos del Departamento de Talento Humano**

**Elaborado por:** Jorge González

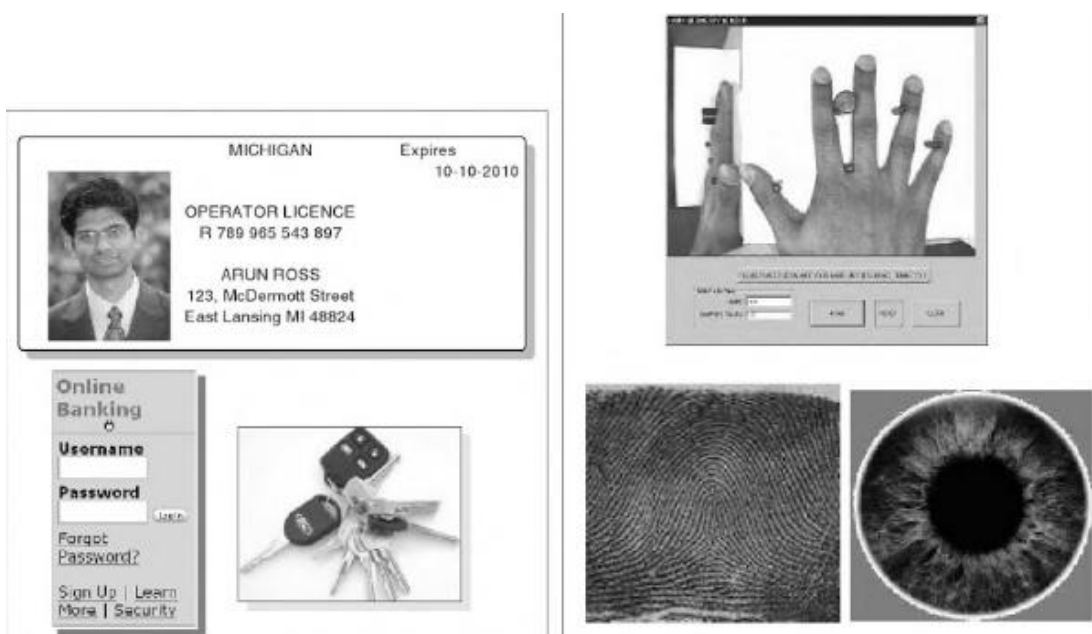
#### **1.7.11. Biometría**

Para Jain et al. (2008), la Biometría es “La ciencia de reconocimiento de identidad de una persona basado en los atributos físicos y de comportamiento del individuo” (p. 1).

A través de la biometría se reconoce la identidad de una persona, basándose en las características físicas, así como también en sus gestos, voz, huellas dactilares e iris.

Algunos ejemplos de uso de la biometría, dados por Jain et al. (2008), son: “verificación para uso de recursos de red, acceso a plantas de energía nuclear, ejecución de transacciones financieras remotas o check-in para abordar un vuelo comercial” (p. 1).

El reconocimiento efectivo, hace que las diferentes técnicas robustas, que se pueden aplicar, sean confiables y permiten brindar seguridad en diferentes procesos, áreas, datos, etc. Algunos ejemplos de su uso son: control de acceso físico de personal a ciertas zonas de una edificación, control de acceso a un equipo de cómputo, registro de ingreso y salida de personal para control de asistencia, etc.



**Figura: 1.2. Esquema de Autenticación.**

**Fuente:** Handbook of Biometrics, 2

#### **1.7.11.1. Operación de un Sistema Biometría**

Al respecto Jain et al. (2008), hace un pronunciamiento del que se comprende que, un sistema biométrico, es un sistema de reconocimiento de modelos, que adquiere información biométrica de una persona, este conjunto de particularidades, son almacenadas en una base de datos, para en algún otro momento ser comparadas y ejecutar una acción basada en los resultados de cotejar los datos. De forma general, un sistema de este tipo, se va a componer de cinco módulos: sensor, valoración de calidad, extracción de rasgos, comparación y base de datos.

#### **1.7.12. Reloj Biométrico ET90**

En el manual de usuario, del reloj biométrico, el fabricante Epordo (2013), menciona

##### **Características.**

- Fácil integración con varios sistemas.
- Admite el registro de hasta 3000 huellas digitales y almacenamiento de 60.000 registros.
- Incluye sensor de imagen de la calidad para dedos secos y mojados.

- Soporta una rotación de identificación de 360 grados.
- Permite ajustar la distorsión de la imagen asegurando la concordancia de huellas dactilares.
- El tiempo de respuesta está alrededor de los 0,7 segundos.
- ET-90 soporta múltiples idiomas.

### Método de Registro

- Huella Digital.
- Password.
- Tarjeta (opcional) (p. 5).



**Figura: 1.3.** Equipo Epordo ET90

Elaborado por: Epordo (2013)

La información del reloj biométrico dada por el fabricante, lo convierte en una buena opción debido a su adaptabilidad a usarse a través de otros sistemas, la información se puede extraer a archivos planos que pueden ser cargados en cualquier base de datos para hacer uso de la información y gestionarla de

la forma en que sea necesaria. Las marcaciones pueden ser receptadas de diversas formas por el dispositivo, para ofrecer facilidades a todo tipo de personas, incluso en aquellos casos en los que, por discapacidad, no puedan registrarse a través de huella digital.

## CAPITULO II

### METODOLOGÍA

#### 2.1. Metodología de Desarrollo

La metodología FDD, según Palmer y Felsing (2002), fue concebida en 1997 y aparece publicada por primera vez, en 1999, en una obra de Peter Coad, Eric Lefebvre y Jeff DeLuca, en los siguientes años, este equipo de personas realiza publicaciones en las que se detalla de mejor manera la aplicación de la metodología, en el sitio [www.nebulon.com/fdd/](http://www.nebulon.com/fdd/). El sitio se denomina la casa de FDD.

Para Letelier y Sánchez (2003), la Metodología FDD: “Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en la fase de diseño e implementación del sistema partiendo de una lista de características que debe reunir el *software*” (p. 7).

La metodología FDD, en español Desarrollo Basado en Funciones ó conocida también como Desarrollo Conducido por Características, es entonces una herramienta para desarrollos ágiles, con entregables a corto plazo y monitoreo constante.

## **Particularidades**

Del sitio, *Feature Driven Development* (s.f.), se resumen las siguientes características,

- Se esfuerza para mantener la calidad, a lo largo de todas las etapas del proyecto, a través de un monitoreo constante.
- Apoya para evitar y solucionar situaciones, como el sobrepasar el presupuesto, errores en el *software* o que los entregables sean menos de lo solicitado por el usuario.
- Define etapas de ejecución cortas, cada dos semanas, con la finalidad de obtener resultados continuos y tangibles.
- Se basa en un proceso repetitivo, que desarrolla un programa funcional que el cliente puede probar y monitorear.
- Define claramente entregables y formas de evaluación del avance del proyecto.
- Insiste, en la manera en que se ejecutan las fases de diseño y construcción.

### **2.1.1. Roles**

Los roles según Palmer y Felsing (2002), se clasifican, en tres categorías, que se aplicarán según el tamaño del proyecto:

- Roles Claves

- Roles de Soporte
- Roles adicionales

Un miembro del equipo de trabajo, pueden tener asignados varios roles.

### 2.1.1.1. Roles Claves

ROL	DESCRIPCIÓN
<b>Gerente de Proyecto</b>	Persona que gerencia el proyecto, toma las decisiones, establece la visión, asigna el personal, presupuesto y realiza el seguimiento.
<b>Arquitecto jefe</b>	Según la magnitud del proyecto puede dividirse en, arquitecto de dominio y arquitecto técnico, se encargan del diseño general del sistema, además se encargan de especificar y aclarar aspectos técnicos.
<b>Gerente de Desarrollo</b>	Se encarga del seguimiento de las actividades de desarrollo, resuelve los inconvenientes de asignación de recursos, puede combinarse en un solo rol con los dos anteriores
<b>Programador jefe</b>	Desarrollador con experticia, se involucra en el análisis del requerimiento y selecciona los rasgos a desarrollar. Líder de un grupo de desarrolladores
<b>Propietarios de clases</b>	El seguimiento a los mismos es ejecutado por el Programador jefe, son desarrolladores individuales
<b>Experto de dominio</b>	Usuario clave, Sponsor, cliente, en el que se apoyan los desarrolladores para levantar el requerimiento

**Tabla: 2.1. Roles Clave**

**Fuente:** Palmer y Felsing (2002)

### 2.1.1.2. Roles de Soporte

ROL	DESCRIPCIÓN
<b>Administrador de entrega</b>	Reporta al Administrador del proyecto, acerca del avance global del desarrollo, según la información que emite cada programador jefe
<b>Gurú de lenguaje</b>	Su conocimiento en lenguajes de programación y TIC's, son consistentes, se encarga de solventar dudas y dar soporte con su ingenio a todo el proyecto
<b>Ingeniero de construcción</b>	Es el responsable de controlar las versiones de los desarrollos y de publicar la documentación
<b>Desarrollador de Herramientas</b>	Se encarga de elaborar herramientas ad-hoc
<b>Administrador del sistema</b>	Vigila el ambiente de trabajo y ejecuta la salida en vivo del sistema

**Tabla: 2.2 Roles de Soporte**  
Fuente: Palmer y Felsing (2002)

### Roles Adicionales

ROL	DESCRIPCIÓN
<b>Verificadores, responsables del despliegue y encargados de documentar</b>	Estos roles pueden ser asignados como adicionales a cualquier miembro del equipo

**Tabla: 2.3 Roles Adicionales**  
Fuente: Palmer y Felsing (2002)

## 2.1.2. Fases de ejecución

Desde su aparición, está compuesta de cinco fases, que dan mayor importancia al diseño y no tanto a la documentación, las fases se muestran en el siguiente gráfico:



**Figura: 2.1. Fases de la metodología FDD**

**Fuente:** Palmer y Felsing (2002)

**Elaborado por:** Jorge González

### 2.1.2.1. FASE 1: Desarrollar un modelo global

Al iniciar el proyecto, es necesario determinar la visión, objetivos, requerimientos de forma general y delimitar el alcance. Una vez creado el modelo global, el equipo de desarrollo, construye un esqueleto del modelo, que más adelante es fraccionando en áreas específicas, en cada una de las cuales se asignan los roles del personal. Lo necesario en esta fase, es que todos los miembros tengan un conocimiento claro del proyecto.

En el sitio oficial, Nebulon.com (s. f.), se informa que, como modelo global, se genera un informe llamado walkthrough que se presenta a todo el grupo de personas con Roles Claves, el mismo documento es utilizado para documentar cada una de las áreas más pequeñas en las que se divide el proyecto. Además, se generan diagramas de clases y de acuerdo con el tamaño del modelo se puede reforzar esta fase, con diagramas de secuencia, para mejor comprensión.

#### **2.1.2.1.1. Ejemplo**

La información descrita a continuación, se obtiene de entrevistas con el Sponsor del Proyecto y el Experto del Dominio, El Sponsor del Proyecto, por lo general es un Gerente de alto nivel o Directivo y el Experto del Dominio está representado por la Jefatura que ejecuta el proceso para el que se va a desarrollar el *software*. La información contenida en el Walkthrough, se registra durante la entrevista y debe tener un lenguaje natural a fin de que sea entendible para el Cliente. El documento, en su primera parte debe poner en

contexto a todo el equipo de desarrollo, acerca del Cliente para el que se va a ejecutar el proyecto y el objetivo principal del *software*.

### **Contexto**

Descripción del cliente para el que se genera el proyecto, misión y visión del mismo. Para mejor entendimiento del requerimiento, se puede añadir el organigrama, los procesos actuales que se van a ver beneficiados por el *software* nuevo.

### **Walkthrough**

Documento, en el que se describirá a breves rasgos el requerimiento, se elabora uno general del proyecto y luego un documento, por cada funcionalidad más pequeña, en la que se crea conveniente dividir el proyecto.

<b>WALKTHROUGH</b>			
Empresa:			
Sponsor del proyecto:			
Fecha inicio:		Fecha fin:	
Versión:			
<b>INFORME</b>			
Equipo de trabajo			
<b>Rol</b>	<b>Tipo de Rol</b>	<b>Nombre</b>	
Área solicitante:			
Objetivo			

Alcance
Antecedentes
Definiciones
Análisis Técnico
Impacto en el negocio
Análisis Económico

**Tabla:** 2.4 Walkthrough Ejemplo  
Elaborado por: Jorge González

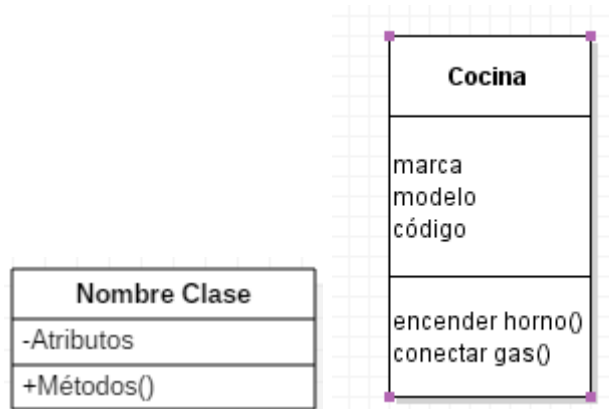
### **Diagrama de Clases**

De la publicación de Gracia (2016), se resume que,

Un diagrama de Clases es útil como diagrama principal para análisis y diseño, está diseñado bajo UML, muestra las entidades del sistema y las relaciones entre ellas, permitiendo visualizar la vista estática del modelo. El diagrama de clases se compone de los siguientes elementos:

- Clase: Atributos, Métodos y Visibilidad

La clase encapsula toda la información de un objeto, sus características y operaciones, se representa por un rectángulo con tres divisiones, en las que se ubica cada componente, como se muestra en el siguiente gráfico



**Figura: 2.2. Estructura de una clase**

**Elaborado por:** Jorge González

Los atributos y métodos, pueden ser de tipo: privado, protegido o público.

- Relaciones: Es la correspondencia que tiene una clase con otra.

Relación	Función	Notación
Asociación	Vínculo entre instancias de clases	
Generalización	Herencia, que receipta una subclase de una súper clase	
Agregación	Representación de elementos que están formados de componentes más pequeños	

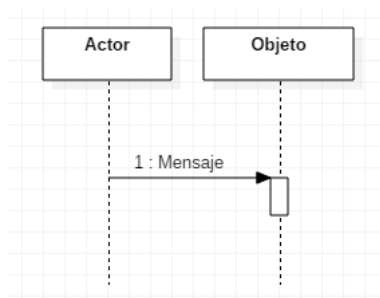
**Tabla: 2.5. Relaciones de Diagrama de Clases**

**Elaborado por:** Jorge González

### **Diagrama de Secuencia**

De la información publicada por Fowler y Scott (1999), se resume que, un diagrama de secuencia, representa la sucesión de interacciones entre las

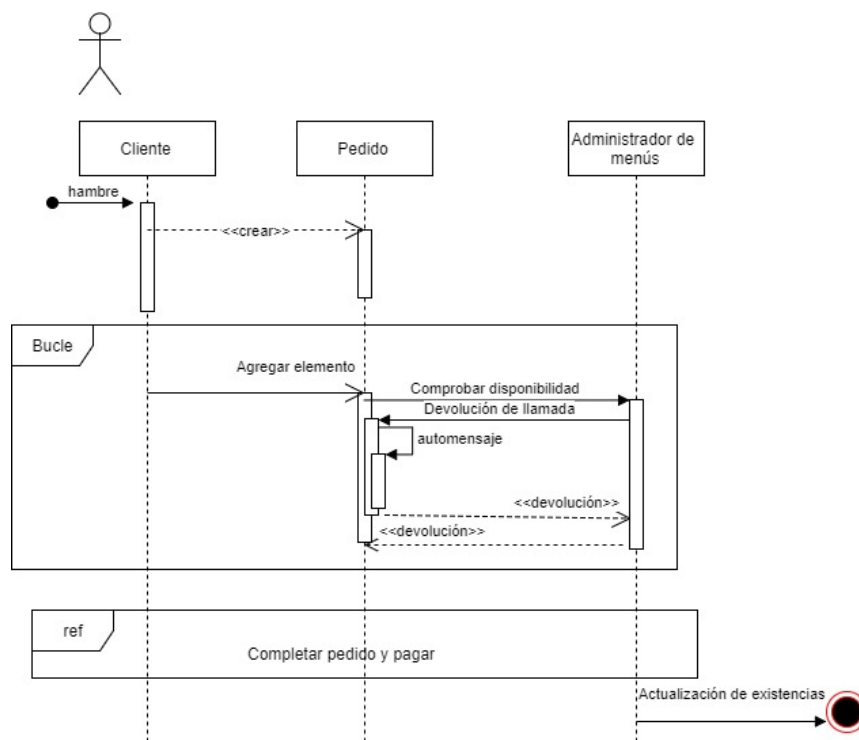
clases, objetos, mensajes y actores, se obtiene a partir del Diagrama de Clases.



**Figura: 2.3. Diagrama de Secuencia**

**Elaborado por:** Jorge González

La dimensión vertical es la línea del tiempo, que avanza hacia la parte inferior del diagrama. La dimensión horizontal contiene los roles de clasificador, cada uno de ellos se representa con una línea vertical conocida como línea de vida. Los mensajes se simbolizan con una flecha, desde la línea de vida de un objeto a otro. Las fechas se sitúan en la línea de vida en orden cronológico.



**Figura: 2.4. Ejemplo de Diagrama de Secuencia**

**Fuente:** <http://msdn.microsoft.com>

### 2.1.2.2. FASE 2: Elaborar una lista de características

Para Pressman (2010), una característica o rasgo, representa el requerimiento del cliente, que se busca expresar de forma detallada a fin de que pueda ser solucionado en dos semanas como máximo.

La característica, se entrega en dos semanas a fin de cumplir con los fundamentos de los Métodos Ágiles, de hacer entregas en corto tiempo y de forma permanente.

En la publicación de Martínez et al. (2015), se indica que en esta fase se elabora una lista con los rasgos recopilados del cliente, en la que se resume las funcionalidades esperadas del sistema, esta lista es evaluada por el cliente y debe ser fraccionada en funcionalidades más pequeñas a fin de que se pueda detallar y lograr un mejor entendimiento del sistema.

En esta metodología, los requerimientos son denominados características y procesados por separado, sin embargo, todas forman parte de una cadena, por tanto, al ir acoplado cada eslabón se va conformando el *software* final.

#### **2.1.2.2.1. Ejemplo**

##### **Lista de características o rasgos**

La lista de características se obtiene de las entrevistas con los Expertos del Dominio y de ser necesario, con la intervención de un Usuario Líder, por ello la importancia de identificar a las personas idóneas. La metodología FDD, no especifica el formato de los registros a utilizar, por tanto, para el diseño de los documentos, se aplicará la experiencia del equipo de trabajo, buenas prácticas y diseños recomendados según el caso de uso y la necesidad.

FUNCIONES VALORADAS POR EL CLIENTE			
Fecha:		Hora inicio:	Hora Fin:
CLASE			
Experto del Dominio			
Usuario Líder			
LISTA DE CARACTERÍSTICAS			

**Tabla:** 2.6. Lista de Características  
**Elaborado por:** Jorge González

De la tesis, de Goyal (2007) se resume que, las características se describen a través de la secuencia: Acción – resultado – objeto, un ejemplo de ello es: Calcular (acción) el total (resultado) de una venta (objeto). Las características que tienen relación, se agrupan en un set de características, que sigue el modelo: Acción – Objeto, por ejemplo: Haciendo (acción) una venta (objeto). Con los sets a su vez, se crea un súper set, que se nombra como: Gestión – Objeto, para este caso Gestión de Ventas.

### **2.1.2.3. FASE 3: Plan por características**

En esta fase, Palmer y Felsing (2002), indican que se debe formar una Secuencia de Desarrollo, con la lista de características, organizándolas de forma sucesiva según su complejidad, el orden en que deben ser desarrollados y de acuerdo a la dependencia de unas con otras, esta tarea es realizada por el administrador del proyecto, administrador de desarrollo y los jefes de programadores. Una vez determinada, la secuencia de las características, se procede con la elaboración del documento de planificación, estableciendo la fecha de inicio y fin y el jefe de desarrollo responsable de cada característica, este documento va a ser útil para la evaluación del avance de actividades. También, se revisa y de ser necesario se hace cambios al diagrama de clases y se asigna el equipo propietario de cada clase.

### 2.1.2.3.1. Ejemplo

#### Secuencia de Desarrollo

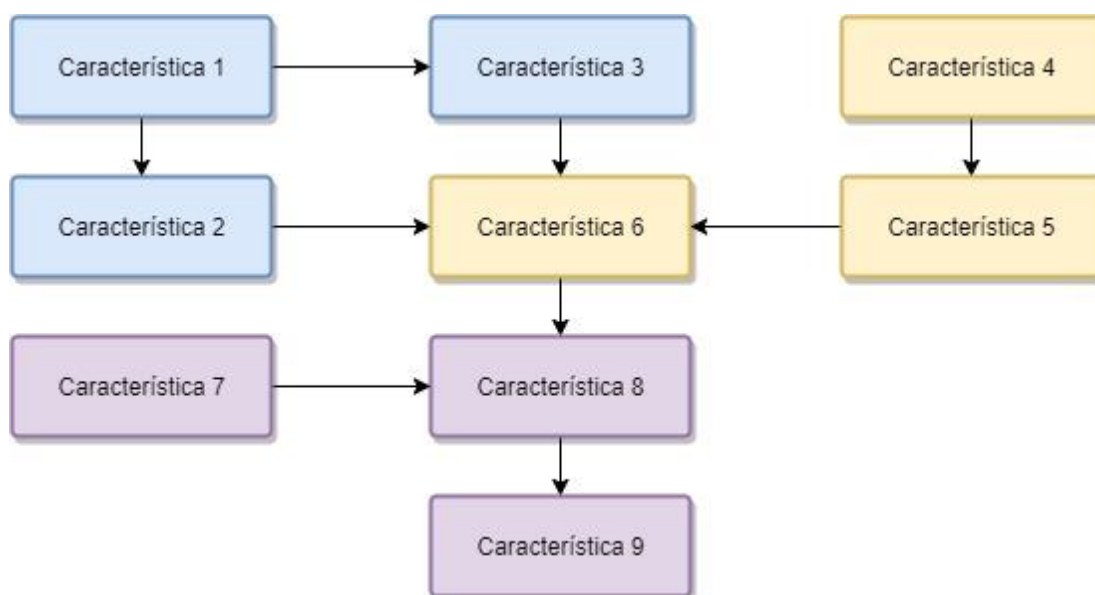
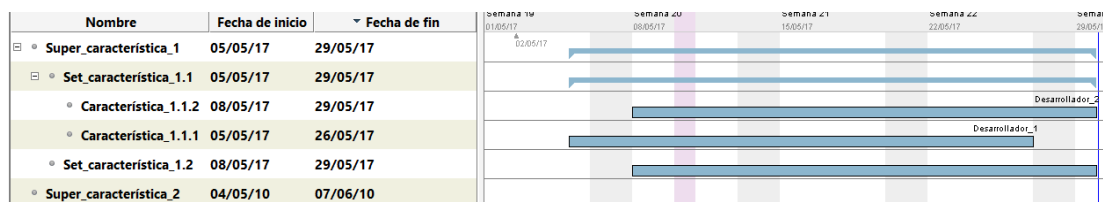


Figura: 2.5. Secuencia de Desarrollo

Elaborado por: Jorge González

#### Planificador de Actividades

Para representar la planificación se puede hacer uso del Diagrama de Gantt, la metodología no establece una herramienta específica, únicamente indica que se debe buscar una alternativa para hacer el seguimiento de las actividades.



#### 2.1.2.4. FASE 4: Diseño por características

Palmer y Felsing (2009) indican que en esta fase los responsables son, el programador jefe y los propietarios de clases, que hacen el seguimiento por paquetes, apoyándose en la arquitectura técnica. En este paso el programador en jefe, reasigna cada una de las características del paquete, que le fue asignado, a los programadores que a su vez se convierten en propietarios de clases. El programador jefe y programadores, revisan los walkthrough, diagramas de clase y secuencia, lista de características y de ser necesario solicitan que se los redefina o se amplíe información. En la Fase 1 no es obligatorio realizar el diagrama de secuencia, sin embargo, al analizar la información en la Fase 4, es posible que se necesite del diseño del mismo. En caso de que se requiera, redefinir el walkthrough o las características, deberá hacerse en consenso con el cliente, para ello es necesario realizar reuniones y dejar documentado en una nueva versión del documento, los cambios, a fin de que estos sean el respaldo de los acuerdos y promesas de servicio. La iteración en esta etapa es de máximo dos semanas, a través de un proceso de modelo de desarrollo de *software* iterativo e incremental, que se utilizan entre la Fase 4 y la Fase 5, en actividades de diseño, codificación, pruebas, integración y revisión de código.

#### **2.1.2.5. FASE 5: Desarrollo por características**

En este paso se deberá, implementar las clases y métodos, realizar inspecciones al código, ejecutar pruebas unitarias e integrales y finalmente la integración.

Cuando Palmer y Felsing (2009) describen esta fase, explican que los responsables son los programadores jefes y propietarios de clases, cada iteración que incluye a las actividades de la fase 4 y 5, puede transcurrir en un lapso de hasta dos semanas, una vez que una iteración es exitosa, la característica completa pasa a productivo y forma parte del sistema general, caso contrario vuelve a la Fase 4. Dicho proceso iterativo se ejecutará hasta concluir el proyecto

Si una característica, hace uso de varias clases, el propietario de cada clase debe hacer el cambio y el desarrollo que asegure la entrega de la característica.

## CAPITULO III

### RESULTADOS

Los resultados que se muestran a continuación se obtienen de la aplicación de las fases de la Metodología FDD en el desarrollo del *software* de Talento Humano, para la Empresa Prodegel.

#### 3.1. FASE 1: Desarrollar un modelo global

##### 3.1.1. Contexto



La Productora de Gelatina Ecuatoriana S.A. (PRODEGEL S.A.), forma parte del grupo GELCO Internacional, esta empresa se dedica a la producción de gelatina, para la industria alimentaria, farmacéutica y fotográfica. El proyecto a desarrollar, apalanca sus valores de Innovación, Compromiso y Responsabilidad, pues desean implementar una herramienta de apoyo, para

el Departamento de Talento Humano, que sea útil para mejorar la ejecución de varios procesos del Área, tales como el registro de información de trabajadores, control de asistencia, permisos de ausencia con sus respectivos respaldos, descuentos y retenciones judiciales al trabajador, formularios de gastos personales, roles de pago, registro de sueldos mensuales y beneficios sociales, registro de vacaciones, finiquitos y cálculo de utilidades de los trabajadores de la empresa y las compañías prestadoras de servicios. El *software* a desarrollar, también va a servir de medio, para que el empleado, solicite permisos y revise su reporte de asistencia.

### 3.1.1.1. Organigramas

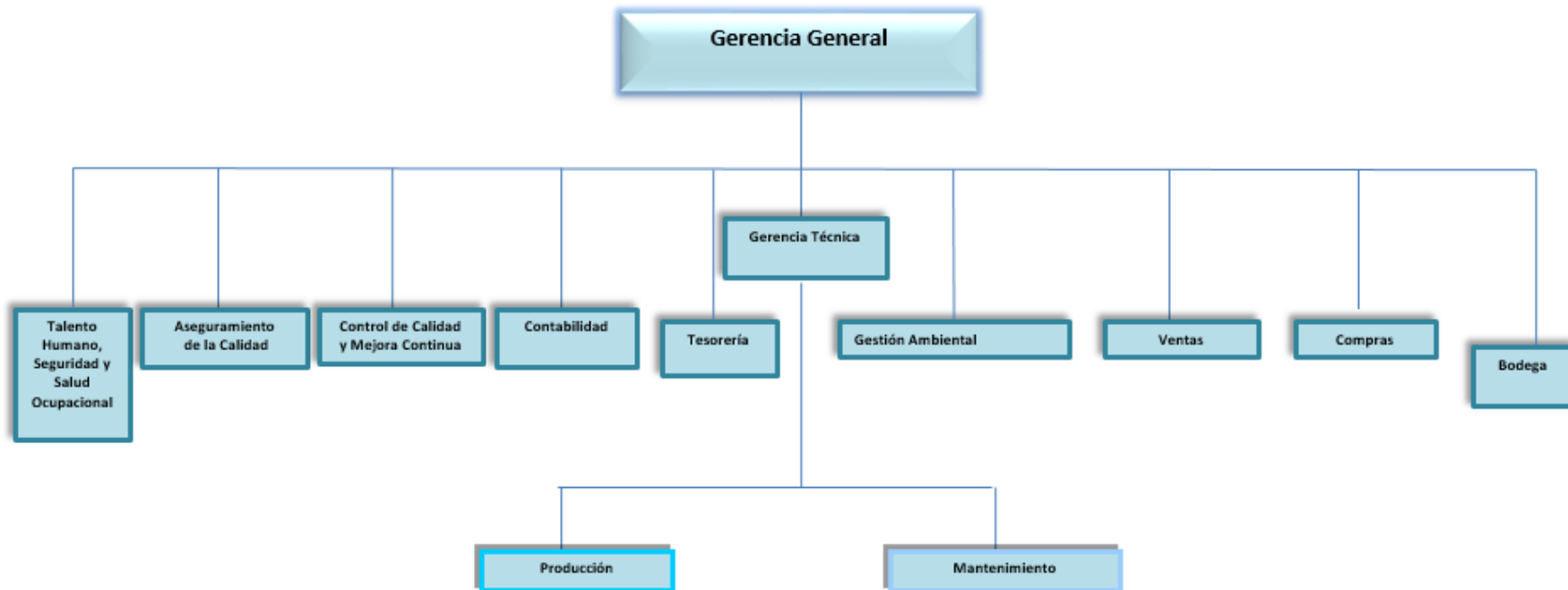
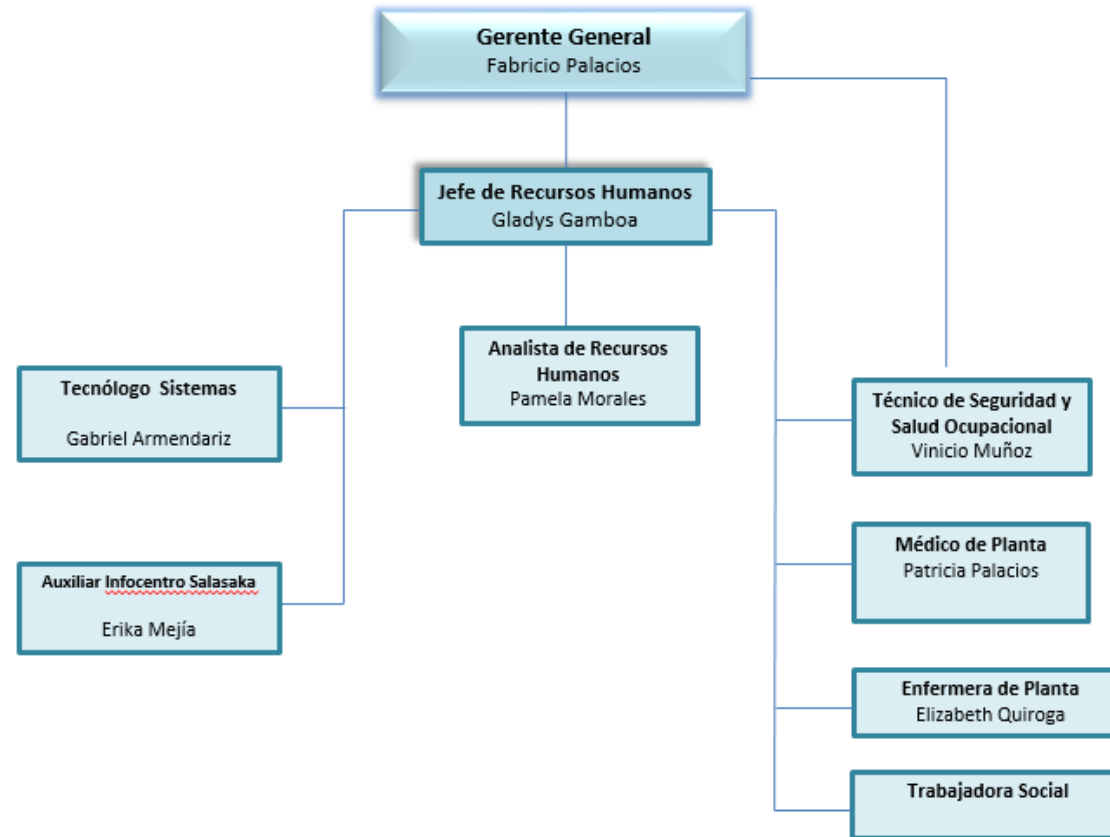


Figura: 3.1. Organigrama General

Fuente: Prodegel S.A.



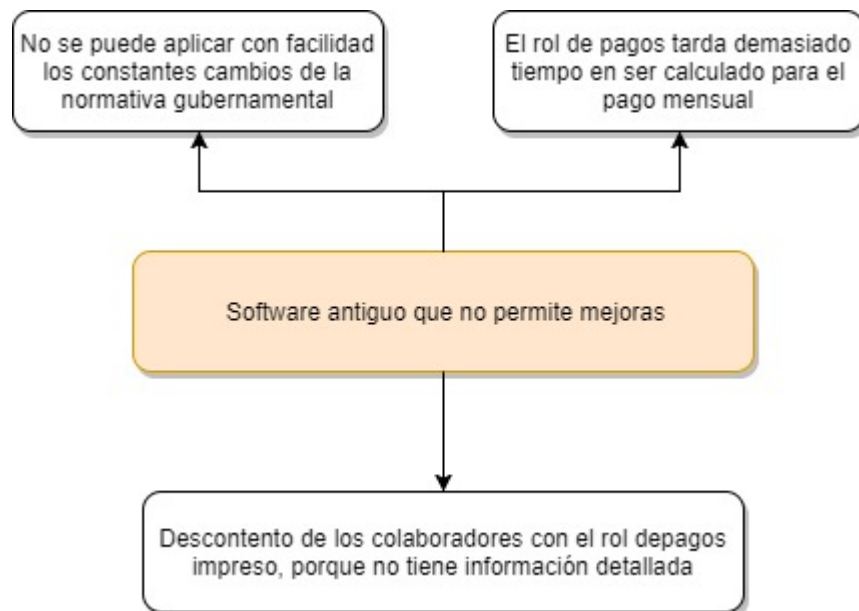
**Figura:** 3.2. Organigrama Talento Humano, Seguridad y Salud Ocupacional

**Fuente:** Prodegel S.A.

### 3.1.2. Walkthrough general

Para generar el walkthrough, se realizó una reunión, con el Gerente General y el equipo de Talento Humano, en la misma se identificó el listado de problemas a través de la Matriz Vester, cuyo resultado se utilizó para generar un árbol de problemas y a partir del mismo se generó el walkthrough.

El acta de reunión y Matriz Vester, se encuentran en el Anexo 1.



**Figura: 3.3** Árbol de Problemas  
**Elaborado por:** Jorge González

<b>WALKTHROUGH</b>			
Empresa:	Productora de Gelatina Ecuatoriana S.A.		
Sponsor del proyecto:	Fabricio Palacios - Gerente General		
Fecha inicio:	29.05.2017	Fecha fin:	29.05.2017
Versión	1		
<b>INFORME</b>			
Equipo de trabajo			
Rol	Tipo de Rol	Nombre	
Administrador del proyecto Arquitecto Jefe	Clave	Jorge González	
Experto de dominio	Clave	Pamela Morales	
Ingeniero de construcción Administrador del sistema	Soporte	Jorge González	
Área solicitante:	Talento Humano		
Objetivo			
<p>Crear un <i>software</i> útil, para la ejecución de los procesos del Área de Talento Humano, que se pueda modificar y expandir en función de las necesidades del negocio.</p>			
Alcance			
<p>El <i>software</i> será utilizado por el personal de Talento Humano, Jefes de Área y Colaboradores, el acceso al módulo de asistencia y permisos, estará disponible desde cualquier lugar a través de Internet, los otros módulos podrán ser accedidos únicamente desde la red local de Prodegel.</p> <p>La aplicación, servirá para: Registro de Empleados, Cálculo de Utilidades, Cálculo de Rol de Pagos, y Control de Asistencia, Permisos y Vacaciones.</p>			
Antecedentes			
<p>* Actualmente el personal del Área de Talento Humano debe invertir gran cantidad de tiempo en la generación del rol de pagos, debido a que no está automatizado y los cálculos de los rubros se realizan en Excel, para luego registrarlos manualmente en el <i>software</i> del que obtienen los reportes y roles de pago individuales.</p> <p>* La información registrada en el rol de pagos impreso, no brinda el detalle requerido para la comprensión de los Colaboradores, debido a que el diseño no es modificable, por tanto, deben asistir constantemente al Departamento de Talento Humano para solicitar información al respecto.</p> <p>* No es posible aplicar con facilidad los cambios de la normativa gubernamental, debido a que el <i>software</i> no es configurable, y se deben aplicar los cambios por fuera, únicamente para registrar en el <i>software</i> los valores calculados.</p> <p>Nota: en los siguientes walkthrough se detallarán los antecedentes según el módulo requerido.</p>			
Definiciones			

<p>Los módulos que formarán parte del <i>software</i> son: Empleados, Nómina, Asistencia y Configuración.  Con la finalidad de que, el <i>software</i> sea amigable y de fácil manejo, se utilizará el desarrollo web, para que los usuarios puedan acoplarse rápidamente.</p>
<b>Análisis Técnico</b>
<p>A fin de no requerir de inversiones altas en TIC's para la implementación del <i>software</i>, se utilizará en gran medida la arquitectura existente, la instalación se realizará en el servidor con sistema operativo Microsoft Windows Server 2012 R2, para el desarrollo se utilizará el lenguaje de programación PHP, los framework Bootstrap y CodeIgniter, manejando perfiles de usuario para delimitar las acciones que cada uno de ellos podrá ejecutar en el sistema, como motor de base de datos se utilizará PostgreSQL, ya que es uno de los más avanzados servidores de base de datos de código libre.</p>
<b>Impacto en el negocio</b>
<ul style="list-style-type: none"> <li>* Mantener información actualizada de empleados y pagos de nómina, para toma de decisiones.</li> <li>* Cumplimiento oportuno y con facilidad, de los cambios de la normativa legal.</li> <li>* Reducción en la inversión de tiempo del personal del Área de Talento Humano en la generación del rol de pagos y cálculo de utilidades.</li> <li>* Creación y configuración de rubros, de acuerdo a las necesidades del negocio.</li> <li>* Incremento en la satisfacción del personal, al tener fácil acceso a su detalle de asistencia</li> <li>* Control de la vacaciones y permisos del personal.</li> <li>* Registros de la asistencia para pago de horas extras y aplicación de multas por atrasos, permitiendo la obtención de reportes de fácil comprensión para el usuario.</li> <li>* Aplicación de TIC's en los procesos internos.</li> </ul>
<b>Análisis Económico</b>
No requiere de inversión

**Tabla: 3.7. Walkthrough General**  
**Elaborado por: Jorge González**

### 3.1.3. Walkthrough por módulos

Para definir los siguientes documentos, se realizó una reunión con la Analista de Recursos Humanos, y dependiendo del caso se solicitó la presencia de otras personas del Área y del Gerente General.

El acta de la reunión, se encuentra en el Anexo 2.

## EMPLEADOS

WALKTHROUGH			
Empresa:	Productora de Gelatina Ecuatoriana S.A.		
Sponsor del proyecto:	Fabricio Palacios		
Fecha inicio:	02.06.2017	Fecha fin:	02.06.2017
Versión	1		
INFORME			
Equipo de trabajo			
Rol	Tipo de Rol	Nombre	
Administrador del proyecto Arquitecto Jefe	Clave	Jorge González	
Experto de dominio	Clave	Pamela Morales	
Ingeniero de construcción Administrador del sistema	Soporte	Jorge González	
Jefe de Recursos Humanos	Usuario	Gladys Gamboa	
Área solicitante:	Talento Humano		
Objetivo			
Mantener una base de datos actualizada, de los colaboradores de la Empresa y de los empleados de las Empresas complementarias.			
Alcance			
La base de datos contendrá información a partir de enero de 2018. El acceso al aplicativo debe ser restringido.			
Antecedentes			
<p><b>EMPLEADOS</b></p> <p>La información de empleados, se registra cuando un nuevo colaborador ingresa a la Empresa, y se actualiza el momento que alguno de los datos que se almacenan en el sistema sufre un cambio. Pero de momento el <i>software</i> no dispone de todos los campos para registro de información, que se requieren, esto obliga al personal de Talento Humano a llevar un registro paralelo de datos en Excel, con la información requerida para la generación de informes gerenciales y gubernamentales, por lo que en ocasiones se genera distorsiones entre las bases de datos.</p> <p>La caracterización del proceso de Incorporación del Recurso Humano, está en el Anexo 2.</p> <p>Al contratar a un colaborador, el personal del Área de Talento Humano, genera el contrato en Microsoft Word, crea la persona en el sistema y los datos adicionales los registran en la</p>			

base de datos de Microsoft Excel. Además, para los casos en los que es necesario se incluyen las cargas familiares.

## INFORMES

La generación de informes de personal, demanda gran inversión de tiempo, por lo que la información en ocasiones, no es oportuna.

### Definiciones

a información que se requiere registrar de los colaboradores, es la siguiente:

#### PERSONAS

##### Identificación General

- Código
- Apellidos
- Nombres
- Identificación
- Lugar de Expedición
- Empresa
- Cédula Militar
- Pasaporte
- Correo Electrónico

##### Información Personal

- Fecha de Nacimiento
- Profesión
- Nivel de Educación
- Sexo
- Estado Civil
- Tipo de Sangre
- Peso
- Talla

##### Teléfonos

- Domicilio
- Celular

##### Residencia

- Provincia
- Región
- Calle Principal
- Número
- Transversal
- Sector
- Referencia
- Tipo de Vivienda
- Vive con Familiares

##### Discapacidad

- Tipo
- Porcentaje

**Cuenta Bancaria**

- Banco
- Tipo
- Número

**Persona de Contacto**

- Nombre
- Dirección
- Parentesco
- Teléfono

**Cargas Familiares**

- Nombre del Familiar
- Identificación
- Parentesco
- Fecha de nacimiento
- Fecha de fallecimiento
- Fecha matrimonio
- Sexo
- Teléfono

**CONTRATOS**

- Código
- Fecha Ingreso
- Fecha Salida
- Sueldo
- Empresa
- Tipo de Contrato
- Área
- Departamento
- Cargo
- Tipo de Trabajo
- Motivo de Salida
- Causa de Salida

Los reportes que el sistema, debe generar son:

- Personal activo: Emitirá el listado de empleados activos, en una fecha específica, los datos que debe presentar son: código de empleado, cédula, apellidos, nombres, Fecha Ingreso, Tipo de Contrato, Área, Departamento, Cargo

- Cargas Familiares Detalle: el reporte debe mostrar el detalle de cargas familiares de cada empleado, dentro de un período de tiempo específico, la información que contiene es cédula del empleado, código, apellidos, nombres, apellidos carga familiar, nombres carga familiar y parentesco.

- Cargas Familiares Consolidado: el reporte debe emitir el número de cargas familiares de cada empleado, en un período de tiempo específico, los datos que muestra son:

<p>* Información centralizada, actualizada y oportuna de los colaboradores, para los diferentes entes de control y para las necesidades del negocio, con un ahorro de recursos considerable al poder obtenerlo en cualquier momento de forma instantánea.</p> <p>* Aplicación de TIC´s en los procesos internos, permitiendo una mejor inversión de recursos.</p>
<b>Análisis Económico</b>
No requiere de inversión

**Tabla: 3.8. Walkthrough Empleado**  
**Elaborado por: Jorge González**

## NÓMINA

WALKTHROUGH			
Empresa:	Productora de Gelatina Ecuatoriana S.A.		
Sponsor del proyecto:	Fabricio Palacios - Gerente General		
Fecha inicio:	05.06.2017	Fecha fin:	05.06.2017
Versión	1		
INFORME			
Equipo de trabajo			
Rol	Tipo de Rol	Nombre	
Administrador del proyecto Arquitecto Jefe	Clave	Jorge González	
Experto de dominio	Clave	Pamela Morales	
Ingeniero de construcción Administrador del sistema	Soporte	Jorge González	
Jefe de Recursos Humanos	Usuario	Gladys Gamboa	
Área solicitante:	Talento Humano		
Objetivo			
Generar el rol de pagos desde un <i>software</i> único.			
Alcance			
<p>El módulo de Nómina será utilizado por la Jefe Recursos Humanos y la Analista de Recursos Humanos, permitirá configurar y crear rubros según las necesidades de la Empresa, para el rol de pagos. Los rubros se deben distinguir entre fijos y variables, a fin de cumplir con las diversas opciones de ingresos, egresos y provisiones. La nómina debe estar enlazada con el módulo de para que se pueda cargar: horas extras, permisos, días laborables y atrasos.</p> <p>La información de días laborables, permitirá realizar el cálculo de las utilidades, por lo que en este módulo se insertara esta opción.</p>			
Antecedentes			

<p>Para generar el rol de pagos del mes, se realizan las siguientes actividades</p> <ul style="list-style-type: none"> <li>* En Microsoft Excel se calculan los diversos rubros de los empleados y se genera un archivo plano por cada rubro calculado para el personal activo.</li> <li>* Se carga el archivo de asistencia de los empleados, el proceso se describe en el walkthrough que hace referencia a dicho módulo.</li> <li>* Los archivos planos de los rubros se cargan al <i>software</i>.</li> <li>* Se procesa la nómina en el <i>software</i> para que se calcule el valor a recibir de cada empleado.</li> </ul> <p>En el Anexo 2, se detallan los rubros que se cargan a través de archivo plano.</p> <p>El rol de pagos se genera a partir del día 22 de cada mes y debe estar listo para pagarse el último día laborable del mes en curso.</p>
<b>Definiciones</b>
<p><b>ROL DE PAGOS</b></p> <p>Para el rol de pagos, se debe tener presente, que se genera uno por mes, con el detalle de los empleados y para cada empleado sus ingresos, egresos y provisiones. Los diferentes rubros que se cargan al rol de pagos, deben ser configurables ya sea por persona o de forma global, dependiendo del caso, por ello deben permitir al usuario generarlos y actualizarlos libremente. La información para determinar días laborables, se debe tomar del módulo de Asistencia. Los reportes requeridos son: Rol de Pagos General y Rol de pagos Individual</p> <p><b>UTILIDADES</b></p> <p>El cálculo de utilidades, se realizará en base a los días laborables del rol de pagos, en abril de cada año, de acuerdo a la ley gubernamental. La información de colaboradores, cargas familiares y empleados de Empresas complementarias se tomarán del módulo de Empleados.</p>
<b>Análisis Técnico</b>
<p>Se aplica el mismo análisis técnico del walkthrough general, para todo el proyecto.</p>
<b>Impacto en el negocio</b>
<p><b>ROL DE PAGOS</b></p> <ul style="list-style-type: none"> <li>• Mantener información actualizada de empleados y pagos de nómina, para toma de decisiones</li> <li>• Cumplimiento oportuno y con facilidad, de los cambios de la normativa legal</li> <li>• Reducción en la inversión de tiempo del personal del Área de Talento Humano en la generación del rol de pagos.</li> <li>• Incremento en la satisfacción del personal, al tener información detallada en el rol de pagos.</li> <li>• Independencia en la configuración de rubros.</li> </ul> <p><b>UTILIDADES</b></p> <ul style="list-style-type: none"> <li>• Realizar el cálculo correcto y de forma ágil, reduciendo la inversión de tiempo del personal.</li> <li>• Centralizar la información de los beneficiarios de las utilidades.</li> </ul>
<b>Análisis Económico</b>

No requiere de inversión
--------------------------

**Tabla:** 3.9. Walkthrough Nómina**Elaborado por:** Jorge González

## ASISTENCIA

WALKTHROUGH			
Empresa:	Productora de Gelatina Ecuatoriana S.A.		
Sponsor del proyecto:	Fabricio Palacios - Gerente General		
Fecha inicio:	06.06.2017	Fecha fin:	06.06.2017
Versión	1		
INFORME			
Equipo de trabajo			
Rol	Tipo de Rol	Nombre	
Administrador del proyecto Arquitecto Jefe	Clave	Jorge González	
Experto de dominio	Clave	Pamela Morales	
Ingeniero de construcción Administrador del sistema	Soporte	Jorge González	
Jefe de Recursos Humanos	Usuario	Gladys Gamboa	
Área solicitante:	Talento Humano		
Objetivo			
Desarrollar un módulo en el <i>software</i> para control de la asistencia del personal, que permitan aplicar el reglamento interno de Prodegel, así como la normativa legal vigente en relación a ausentismo, permisos remunerados o no, vacaciones, atrasos, faltas injustificadas, etc.			
Alcance			
<p>* El módulo receptorá los registros de ingresos y salidas del personal, que se generan en los relojes biométricos, con esos datos se calculará las horas extras, atrasos, faltas, vacaciones, permisos y días laborables.</p> <p>* Permitirá a los empleados, el registro de solicitud de permisos a través de un sitio web y la aprobación de los mismos los realizará el Responsable del Área.</p> <p>* Será útil para el registro de las vacaciones de los colaboradores</p> <p>* Se utilizará como respaldo, para la aplicación de sanciones por faltas al reglamento interno.</p>			
Antecedentes			

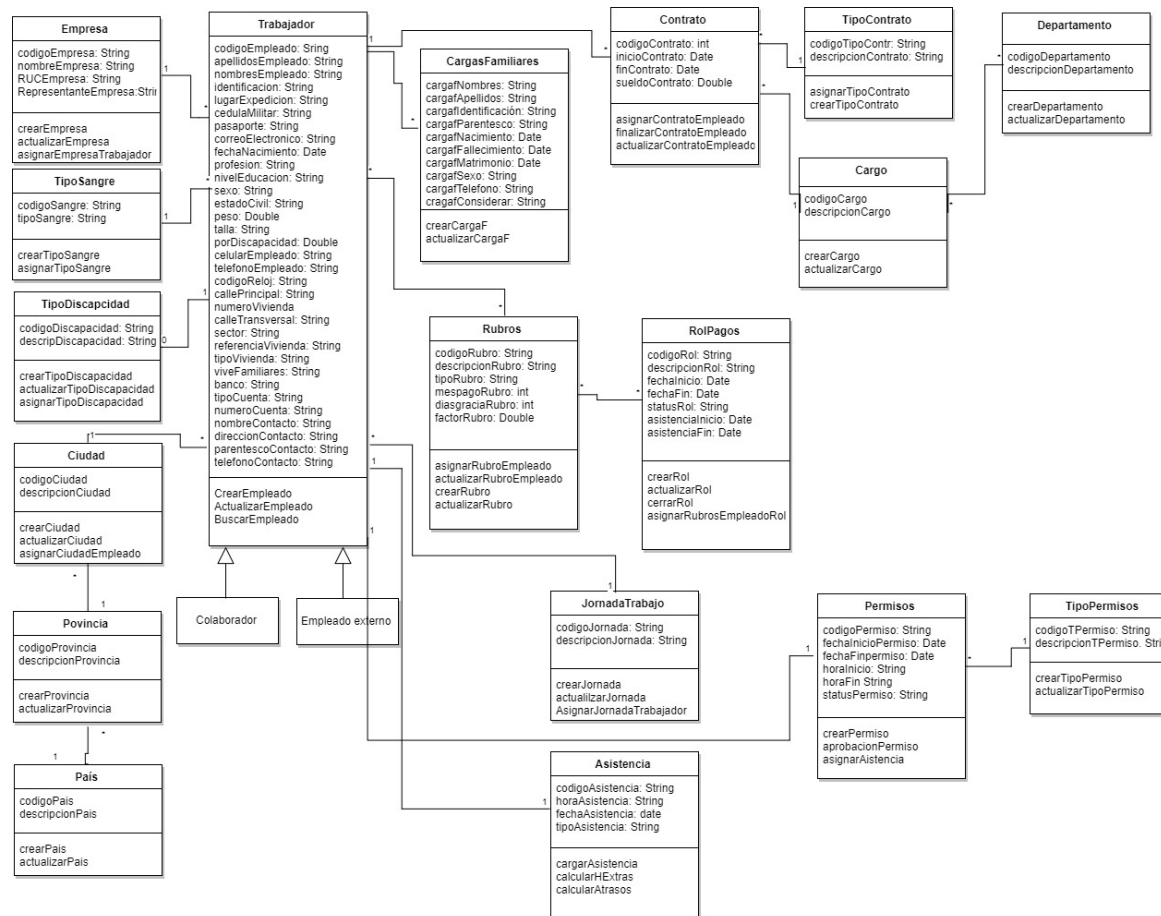
<p>* En la actualidad el personal del Área de Talento Humano, realiza el control de las vacaciones del personal en un archivo de Excel, en el que enlista el histórico de los días de vacación pendiente y vacaciones gozadas de cada empleado, por año de servicio.</p> <p>* No existe ningún control automatizado para permisos, el personal de Talento Humano recibe el listado, de detalle de permisos de cada área para el cierre de nómina, a través de correo electrónico y lo cuadra con el archivo plano que descarga de los relojes biométricos.</p> <p>* Los reporte de atrasos, faltas y horas extras, se obtienen del <i>software</i> del reloj biométrico, el cálculo lo realiza en función de los horarios asignados al personal, con esta información la Asistente de Talento Humano, calcula en Excel los valores a pagar o descontar según el caso y los digita en el <i>software</i> de nómina.</p>
<b>Definiciones</b>
<p>El módulo de Asistencia, debe tener el Registro de Asistencia, Opción de Aprobación de Horas, Ausentismo (permisos, vacaciones) y Solicitudes de Permisos.</p> <p>Los reportes que debe tener son: Ausentismo, Registro de Asistencia, Solicitud de Vacaciones, Anticipo de Vacaciones y permisos.</p> <p>El personal Operativo, trabaja en tres turnos rotativos de 8 horas, de forma continua y el personal Administrativo trabaja en jornada única de lunes a viernes.</p>
<b>Análisis Técnico</b>
<p>A fin de no requerir de inversiones altas en TIC's para la implementación del <i>software</i>, se utilizará en gran medida la arquitectura existente, la instalación se realizará en el servidor con sistema operativo Microsoft Windows Server 2012 R2, para el desarrollo se utilizará el lenguaje de programación PHP y el framework Bootstrap, manejando perfiles de usuario para delimitar las acciones que cada uno de ellos podrá ejecutar en el sistema, como motor de base de datos se utilizará PostgreSQL, ya que es uno de los más avanzados servidores de base de datos de código libre.</p>
<b>Impacto en el negocio</b>
<ul style="list-style-type: none"> <li>* Controlar las vacaciones gozadas y pendientes.</li> <li>* Mantener un respaldo de la asistencia del personal.</li> <li>* Optimizar el gasto en pago de horas extras.</li> <li>* Facilidad de aplicación del reglamento interno de trabajo.</li> </ul>
<b>Análisis Económico</b>
No requiere de inversión

**Tabla: 3.10. Walkthrough Asistencia**  
**Elaborado por: Jorge González**

#### **3.1.4. Diagrama de clases**

Para generar el Diagrama de clases, se necesita de un conocimiento más al detalle, de los requerimientos y procesos a solventar, por lo que se realiza una reunión con el personal del Área de Talento Humano, el detalle de la misma se encuentra en el Anexo 2, con el nombre Acta 003.

El Diagrama de Clases, se generó bajo la perspectiva conceptual



**Figura: 3.4. Diagrama de Clases**  
**Elaborado por: Jorge González**

### 3.2. FASE 2: Elaborar una lista de características

La siguiente lista de características se generó en base a una reunión con el personal del área de Talento Humano, y según las especificaciones del diagrama de clases.

FUNCIONES VALORADAS POR EL CLIENTE					
Fecha:	12.06.2017	Hora inicio:	8:30	Hora Fin:	16:00
Versión	1				
	Experto del Dominio	Gladys Gamboa			
	Usuario Líder	Pamela Morales			
LISTA DE CARACTERÍSTICAS					
Número	Módulo	Descripción	Clase		
1	EMPLEADO	Crear un país en la base de datos	País		
2	EMPLEADO	Actualizar un país en la base de datos	País		
3	EMPLEADO	Crear una provincia	Provincia		
4	EMPLEADO	Actualizar provincias	Provincia		
5	EMPLEADO	Crear ciudades	Ciudad		
6	EMPLEADO	Actualizar ciudades	Ciudad		
7	EMPLEADO	Crear tipos de sangre	Tipos_sangre		
8	EMPLEADO	Actualizar tipos de sangre	Tipos_sangre		
9	EMPLEADO	Crear tipos de trabajador	tipo_trabajador		
10	EMPLEADO	Actualizar tipos de trabajador	tipo_trabajador		
11	EMPLEADO	Crear departamentos de la empresa	Departamentos		
12	EMPLEADO	Actualizar departamentos de la empresa	Departamentos		
13	EMPLEADO	Crear los cargos	Cargos		
14	EMPLEADO	Actualizar los cargos	Cargos		
15	EMPLEADO	Crear el trabajador en el sistema	Trabajador		
16	EMPLEADO	Actualizar el trabajador en el sistema	Trabajador		
17	EMPLEADO	Asignar un contrato al trabajador	Contratos		
18	EMPLEADO	Actualizar contrato de trabajo	Contratos		
19	EMPLEADO	Agregar cargas familiares de trabajador	cargas_familiares		
20	EMPLEADO	Actualizar cargas familiares de trabajador	cargas_familiares		
21	EMPLEADO	Actualizar información de contrato	Contratos		
22	EMPLEADO	Registrar la finalización del contrato	Contratos		
23	NOMINA	Creación de rol de pagos	Roldepagos		
24	NOMINA	Actualización de rol de pagos	Roldepagos		
25	NOMINA	Crear rubros	Rubros		

26	NOMINA	Actualizar rubros	Rubros
27	ASISTENCIA	Obtener los registros de asistencia	Asistencia
28	ASISTENCIA	Crear jornada de trabajo	jornada_trabajo
29	ASISTENCIA	Actualizar jornada de trabajo	jornada_trabajo
30	ASISTENCIA	Asignar jornada de trabajo	jornada_trabajo
31	ASISTENCIA	Crear tipos de permiso	Permisos
32	ASISTENCIA	Actualizar tipos de permiso	Permisos
33	ASISTENCIA	Solicitar permiso	Permisos
34	ASISTENCIA	Aprobación de permiso	Permisos
35	ASISTENCIA	Solicitar vacaciones	Vacaciones
36	ASISTENCIA	Aprobación de vacaciones	Vacaciones

**Tabla:** 3.11. Lista de características  
**Elaborado por:** Jorge González

## **Agrupación por Iteraciones**

### **EMPLEADO**

#### **Iteración 1**

1. Crear países
2. Actualizar países
3. Crear provincias
4. Actualizar provincias
5. Crear ciudades
6. Actualizar ciudades
7. Crear tipos de sangre
8. Actualizar tipos de sangre
9. Crear tipos de trabajador
10. Actualizar tipos de trabajador
11. Crear departamentos de la empresa
12. Actualizar departamentos de la empresa

13. Crear los cargos

14. Actualizar los cargos

### **Iteración 2**

1. Crear el trabajador en el sistema

2. Actualizar el trabajador en el sistema

3. Asignar un contrato al trabajador

4. Actualizar contrato de trabajo

5. Agregar cargas familiares de trabajador

6. Actualizar cargas familiares de trabajador

7. Actualizar información de contrato

8. Registrar la finalización del contrato

### **Iteración 3**

1. Crear rubros

2. Actualizar rubros

## **ASISTENCIA**

### **Iteración 4**

1. Obtener los registros de asistencia

### **Iteración 5**

1. Crear jornada de trabajo

2. Actualizar jornada de trabajo

3. Asignar jornada de trabajo

### **Iteración 6**

1. Crear tipos de permiso

2. Actualizar tipos de permiso

#### **Iteración 7**

1. Solicitar permiso
2. Aprobación de permiso

#### **Iteración 8**

1. Solicitar vacaciones
2. Aprobación de vacaciones

### **NOMINA**

#### **Iteración 9**

1. Creación de rol de pagos
2. Actualización de rol de pagos

#### **Iteración 10**

1. Reporte de trabajadores activos
2. Reporte de trabajadores por departamento
3. Reporte de trabajadores por tipo de contrato
4. Reporte de cargas familiares por trabajador
5. Rol de pagos individual
6. Rol de pagos general

### **3.3. FASE 3: Plan por características**

El desarrollo de cada una de las iteraciones se realizará en un tiempo máximo de 2 semanas, en cada iteración está implícito el desarrollo, pruebas, correcciones y aceptación del cliente.

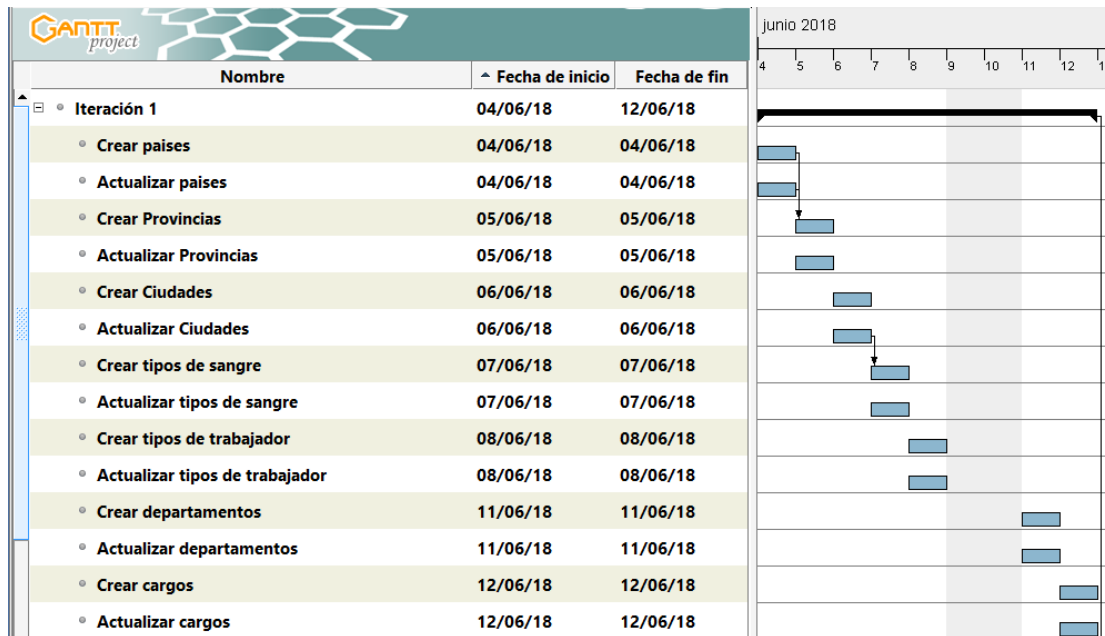


Figura: 3.5. Plan de entrega Iteración 1

Elaborado por: Jorge González



Figura: 3.4. Plan de entrega Iteración 2

Elaborado por: Jorge González



Figura: 3.5. Plan de entrega Iteraciones 3 y 4

Elaborado por: Jorge González



Figura: 3.6. Plan de entrega Iteración 5

Elaborado por: Jorge González

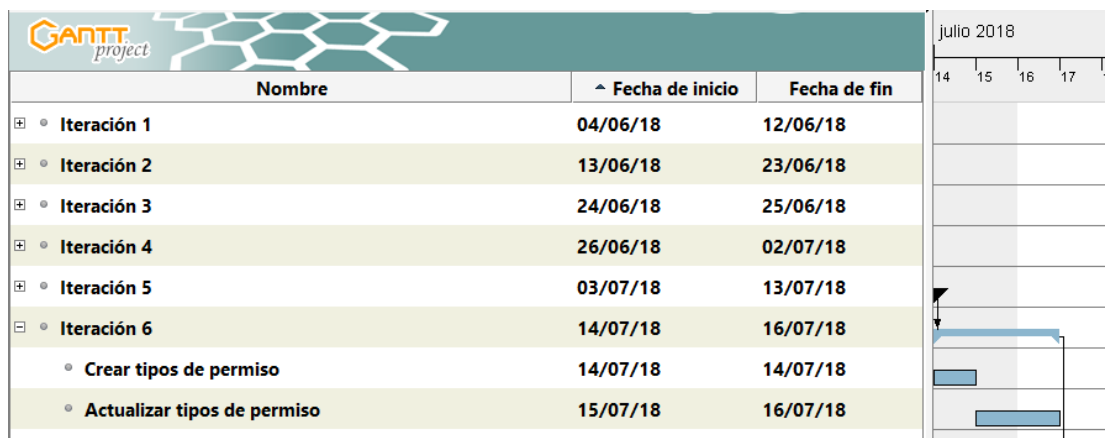
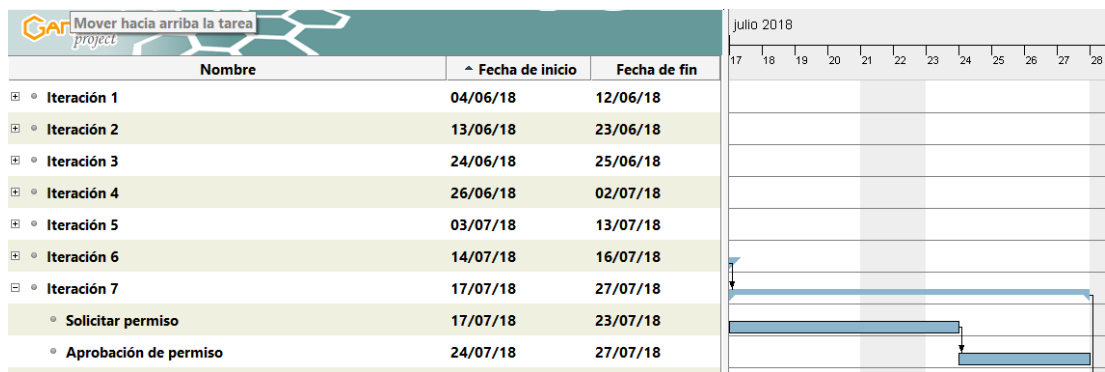


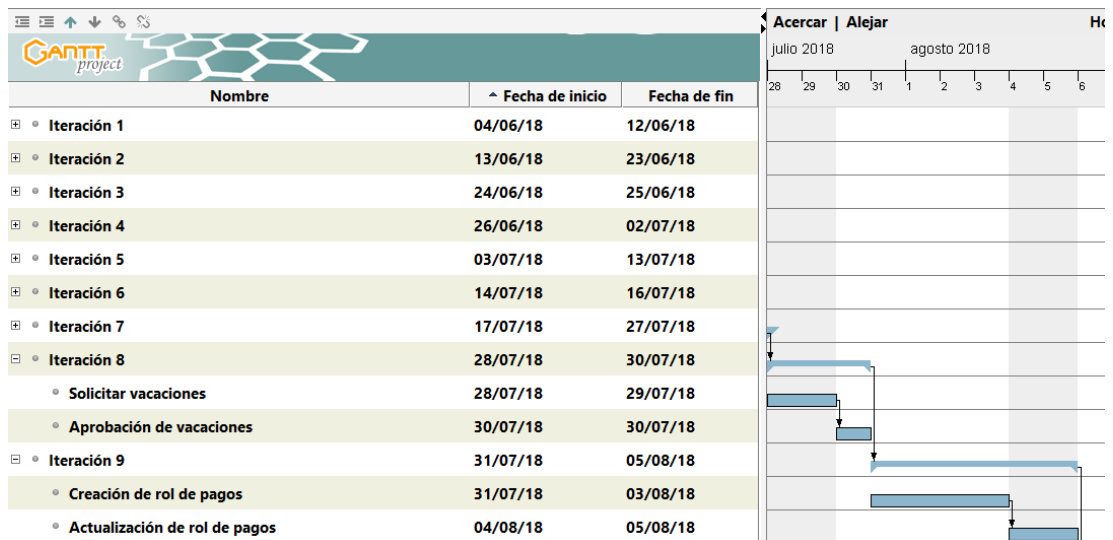
Figura: 3.7. Plan de entrega Iteración 6

Elaborado por: Jorge González



**Figura: 3.8.** Plan de entrega Iteración 7

**Elaborado por:** Jorge González



**Figura: 3.9.** Plan de entrega Iteraciones 8 y 9

**Elaborado por:** Jorge González

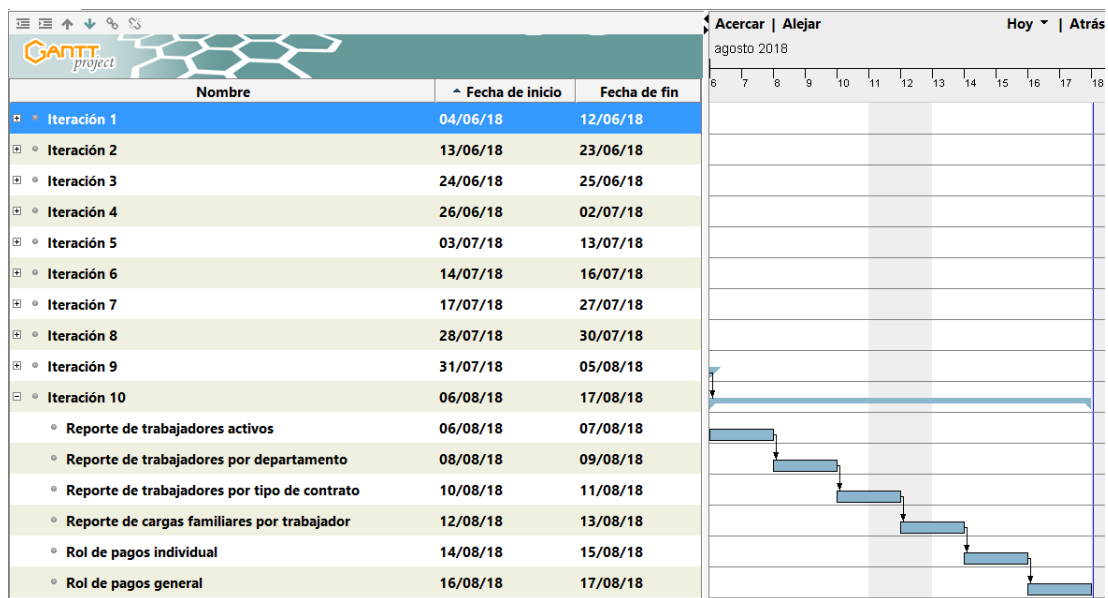


Figura: 3.110. Plan de entrega Iteración 10

Elaborado por: Jorge González

Al ser una metodología iterativa se documentará las fases 4 y 5 por iteración

### 3.4. FASE 4: Diseño por características

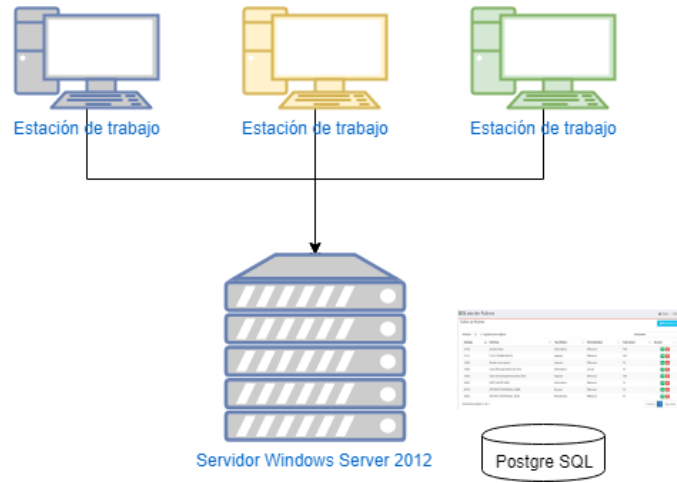
#### 3.4.1. Arquitectura Lógica



Figura: 3.13. Arquitectura Lógica

Elaborado por: Jorge González

#### 3.4.2. Arquitectura Física



**Figura: 3.14.** Arquitectura Física

**Elaborado por:** Jorge González

### 3.4.3. Diseño de la base de datos

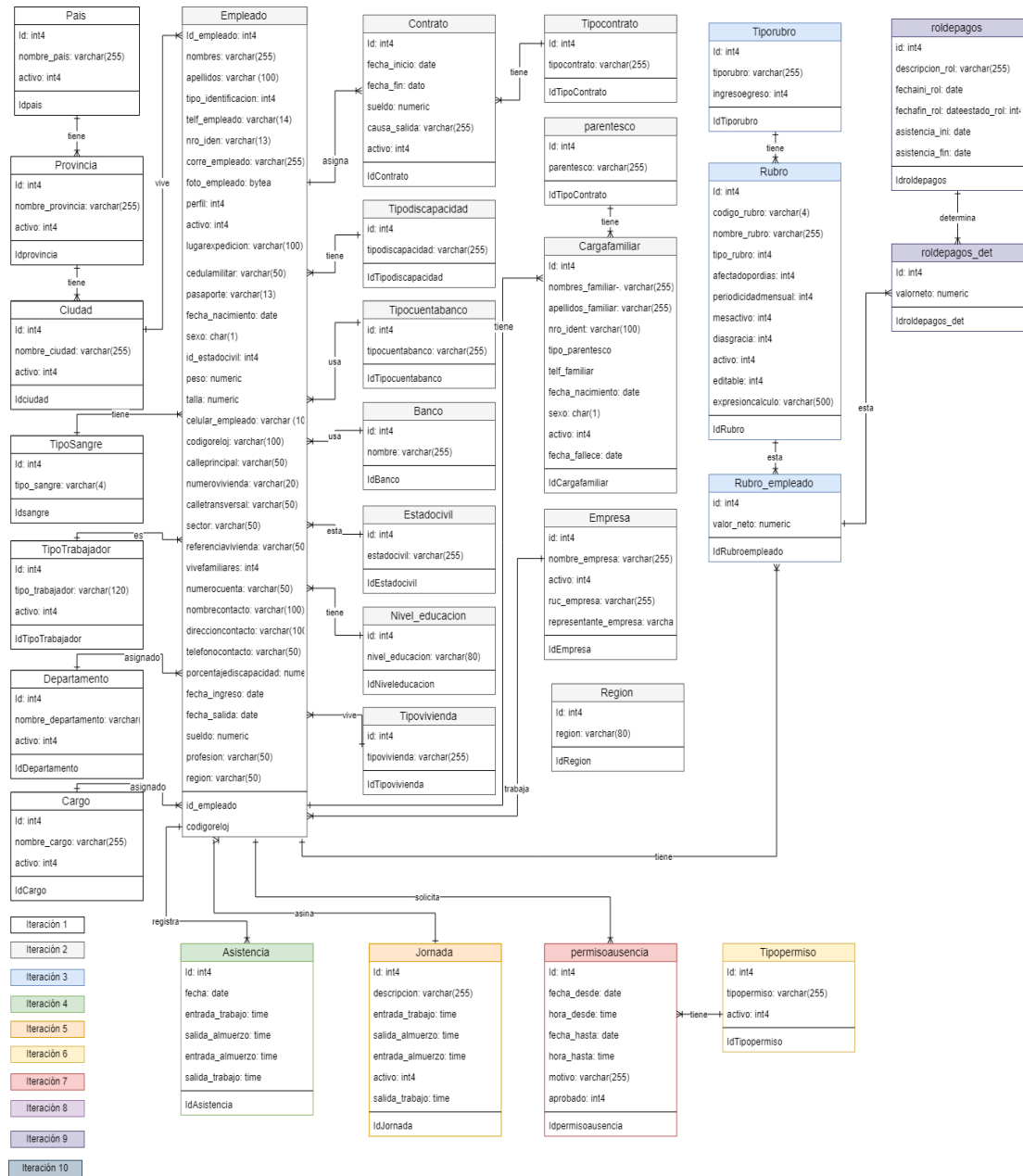


Figura: 3.15. Modelo Lógico de Base de Datos

Elaborado por: Jorge González

### **3.5. FASE 5: Desarrollo por características**

En el ANEXO 3 se muestra el código fuente de las clases creadas, a partir de las iteraciones.

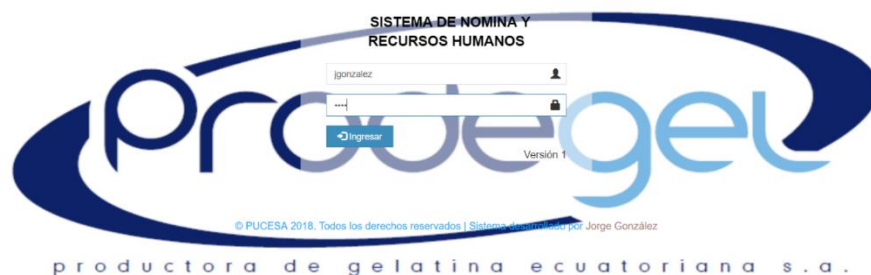
## CAPITULO IV

### ANÁLISIS Y VALIDACIÓN DE RESULTADOS

#### 4.1. Manual

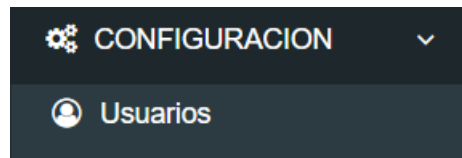
La primera vez que se ejecute el sistema, por defecto se creará en la base de datos el usuario admin, clave admin.

##### 4.1.1. Creación de usuario Administrador



1. Doble clic en el acceso directo del escritorio “Prodegel RRHH” o en un navegador web ingrese en la dirección <http://192.168.1.100/prodegelrrhh/>.
2. Digitar su nombre de usuario.
3. Digitar contraseña.

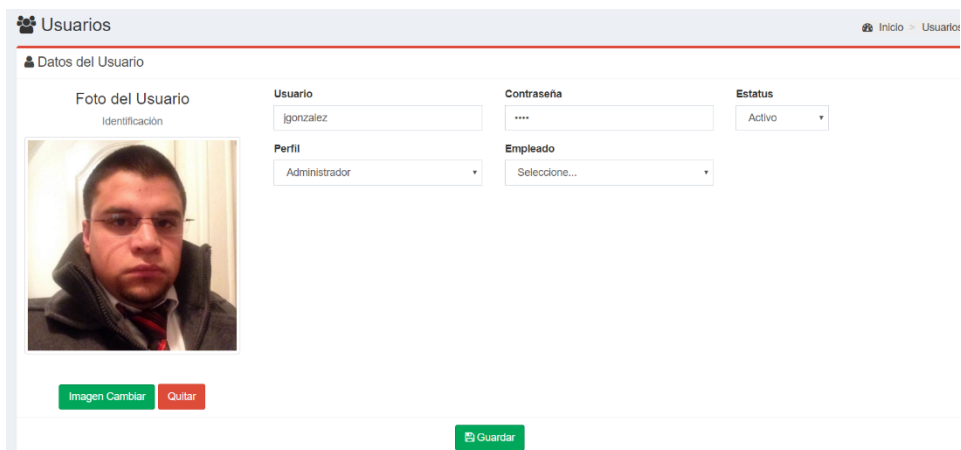
4. Clic en el botón de Ingresar.
5. Una vez que se ha ingresado al sistema, seleccionar la opción de usuarios dentro del menú configuración.



6. Clic sobre



7. Llenar los datos del nuevo usuario administrador

A screenshot of a web application interface for user management. The title is 'Usuarios' with a breadcrumb 'Inicio > Usuarios'. Below the title is a section 'Datos del Usuario' containing a form. The form has several fields: 'Foto del Usuario' with a placeholder 'Identificación' and a photo of a man; 'Usuario' with the value 'jgonzalez'; 'Contraseña' with masked characters '\*\*\*\*'; 'Estatus' with a dropdown menu showing 'Activo'; 'Perfil' with a dropdown menu showing 'Administrador'; and 'Empleado' with a dropdown menu showing 'Seleccione...'. At the bottom of the form are buttons for 'Imagen Cambiar', 'Quitar', and 'Guardar'.

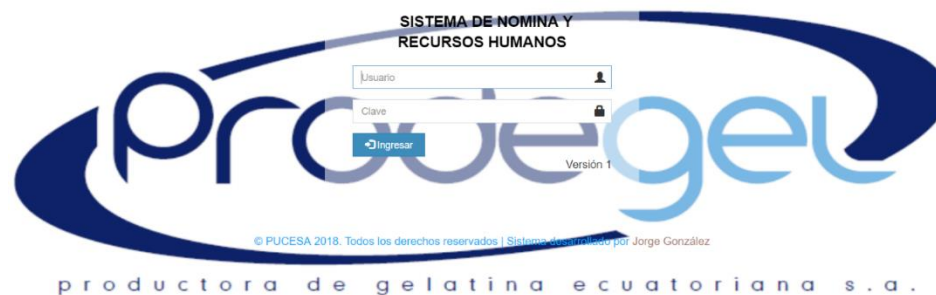
8. Clic en



El nuevo usuario administrador será quién configurará todos los parámetros del sistema de acuerdo al manual de usuario.

#### 4.1.2. Ingreso al sistema

1. Doble clic en el acceso directo del escritorio “Prodegel RRHH” o en un navegador web ingrese en la dirección <http://192.168.1.119/prodegelrrhh/trunk/inicio>.
2. Digitar su nombre de usuario.
3. Digitar contraseña.
4. Clic en el botón de Ingresar.



5. Digitar su nombre de usuario.
6. Digitar contraseña.
7. Clic en el botón de Ingresar.

Una vez que se haya iniciado sesión, se mostrará la pantalla principal del sistema, en la que se mostrará el nombre del usuario activo, así como todas las opciones del menú.



## Opciones de menú principal

**EMPLEADOS:** Crear y actualizar departamentos, cargos, tipo de trabajador, empresas y empleados.

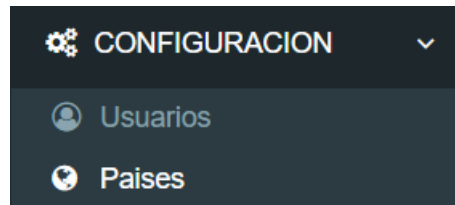
**NOMINA:** Crear y actualizar todos los rubros que recibirán los trabajadores, así como los roles de pago generales e individuales.


**ASISTENCIA:** Es útil para generar crear y actualizar jornadas u horarios de trabajo, y visualizar los registros de asistencia por fecha o por trabajador.

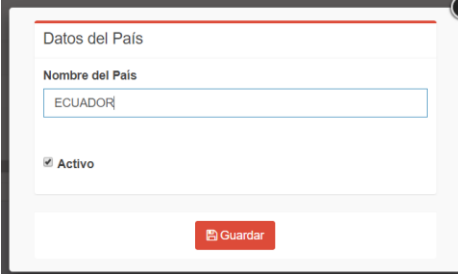
**CONFIGURACIÓN:** Crear y actualizar países, provincias, ciudades, usuarios, correo electrónico, bancos, parámetros generales.

### 4.1.3. Creación de países

1. Seleccione la opción de países dentro del menú de configuración



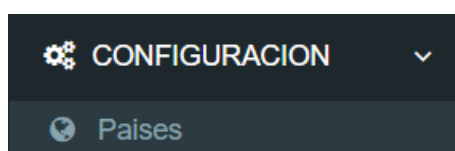
2. Clic sobre el botón 
3. Escribir el nombre del país y verificar que la opción de activo esté marcada.



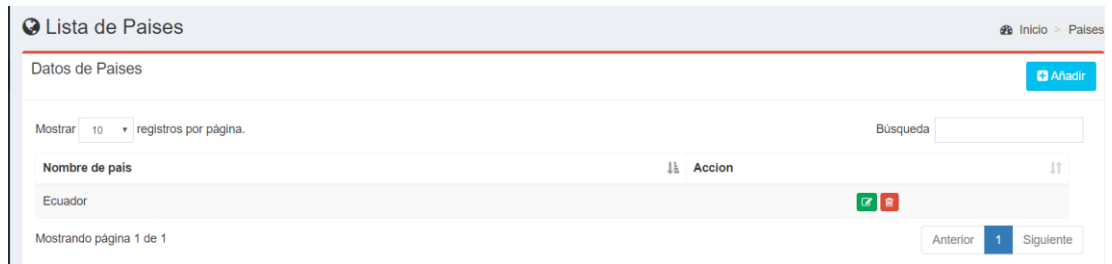
4. Clic en 


### 4.1.4. Actualización de países

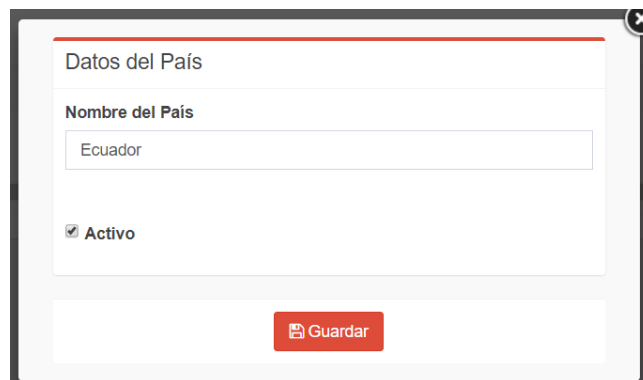
1. Seleccione la opción de países dentro del menú de configuración.



2. Se cargará la lista de todos los países creados en el sistema.



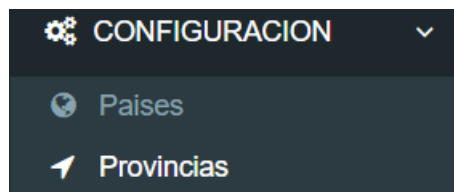
3. Dar clic sobre el botón  en la línea correspondiente al país que se desea editar.
4. Editar la información del país.




5. Clic en 

#### 4.1.5. Creación de provincias

1. Seleccionar la opción de provincias dentro del menú configuración.



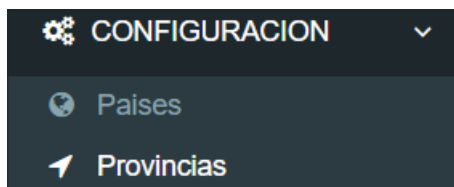
2. Clic sobre el botón 
3. Escribir el nombre de la provincia, seleccionar el país al que corresponde y verificar que la opción de activo se encuentre marcada.

A modal window titled 'Datos de Provincia' with a close button in the top right corner. It contains three input fields: 'Nombre de Provincia' with the text 'Tungurahua', 'País' with a dropdown menu showing 'Ecuador', and a checked checkbox for 'Activo'. At the bottom right is a red button labeled 'Guardar Provincia'.

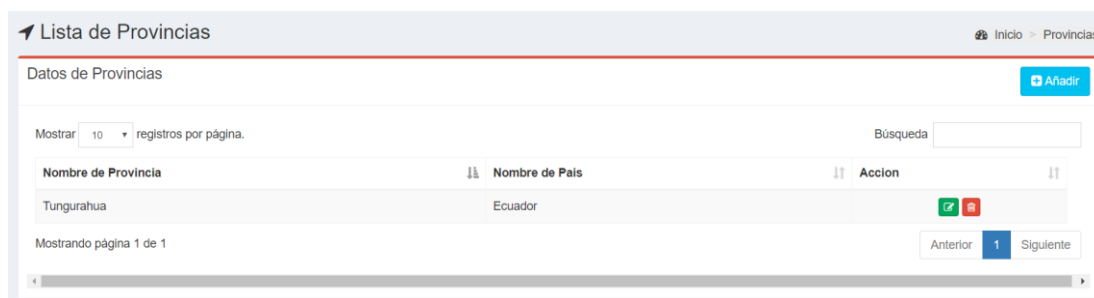
4. Clic en 


#### 4.1.6. Editar una provincia

1. Seleccionar la opción de provincias dentro del menú configuración.



2. Se cargará la información de las provincias que se hayan registrado en el sistema.



3. Clic sobre el botón  en la línea correspondiente a la provincia que se desea editar.

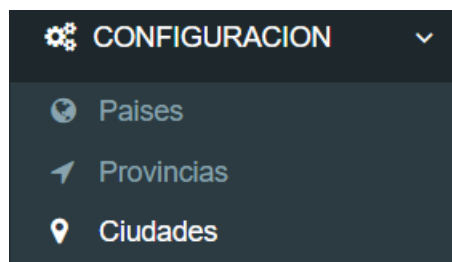



4. Cambiar el nombre de la provincia o el país al que pertenece.

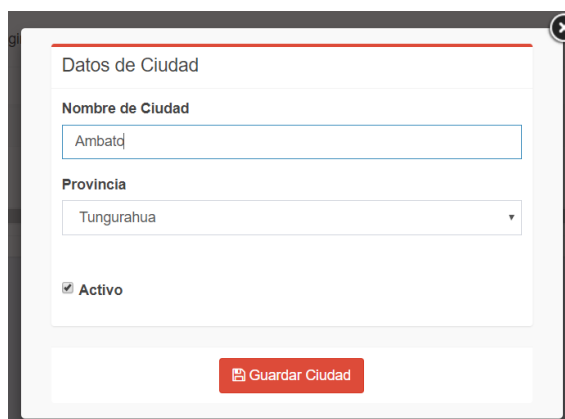
5. Clic en 

#### 4.1.7. Creación de ciudades

1. Seleccionar la opción de ciudades dentro del menú de configuración.



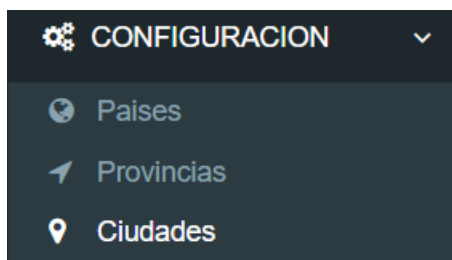
2. Clic sobre 
3. Escribir el nombre de la ciudad, seleccionar la provincia a la que pertenece y verificar que la opción de activo esté marcada.

A light-colored form titled 'Datos de Ciudad' with a close button in the top right corner. It contains three fields: 'Nombre de Ciudad' with the text 'Ambatd' entered; 'Provincia' with a dropdown menu showing 'Tungurahua'; and a checked checkbox labeled 'Activo'. At the bottom right of the form is a red button labeled 'Guardar Ciudad'.

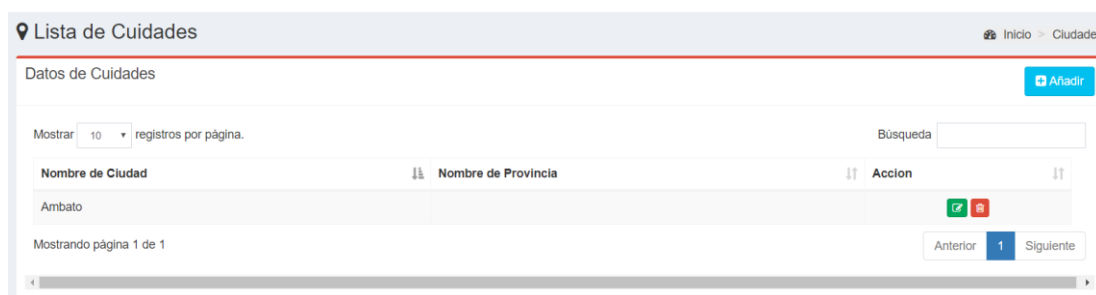
4. Clic en 


#### 4.1.8. Editar ciudades

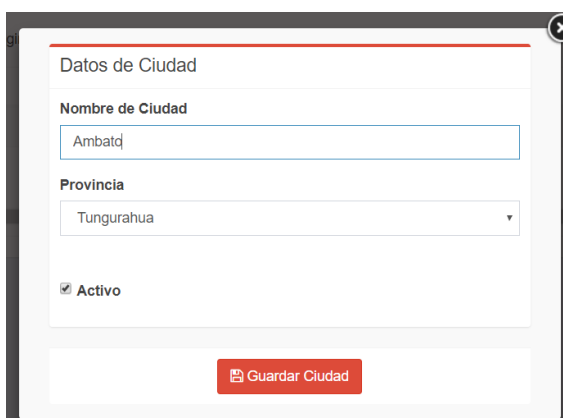
1. Seleccionar la opción de ciudades dentro del menú de configuración.



2. Se cargará la información de todas las ciudades registradas en el sistema



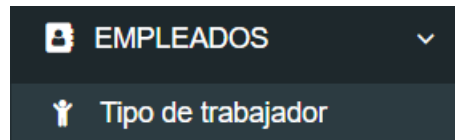
3. Clic sobre el botón  en la línea correspondiente a la ciudad que se desea editar.
4. Cambiar el nombre de la ciudad o la provincia a la que pertenece.



5. Clic en 

#### 4.1.9. Crear tipo de trabajador

1. Seleccionar la opción tipo de trabajador dentro del menú empleados



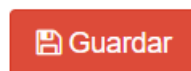
2. Clic sobre



3. Digitar el tipo de trabajador y una descripción del mismo, verificar que el campo activo esté marcado

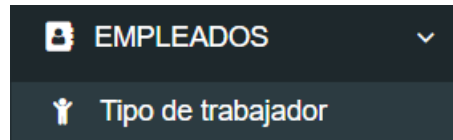
A modal window titled 'Datos de Tipo de Empleado' with a close button in the top right corner. It contains three input fields: 'Tipo de Empleado' with the value 'Obrero', 'Descripcion' with the value 'Obrero de producción', and a checked checkbox labeled 'Activo'. At the bottom center is a red 'Guardar' button with a white floppy disk icon.

4. Clic en

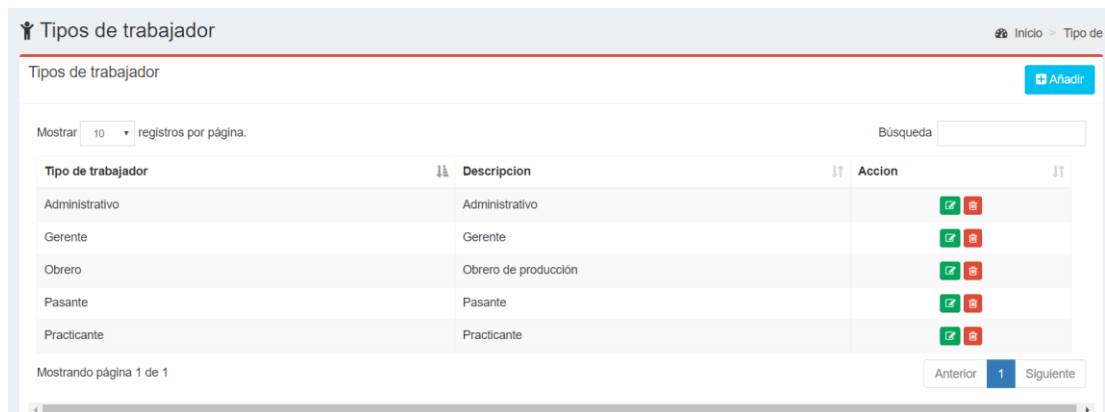


#### 4.1.10. Editar tipo de trabajador

1. Seleccionar la opción tipo de trabajador dentro del menú empleados













2. Se cargará la información de los tipos de trabajador creados en el sistema.




Tipos de trabajador

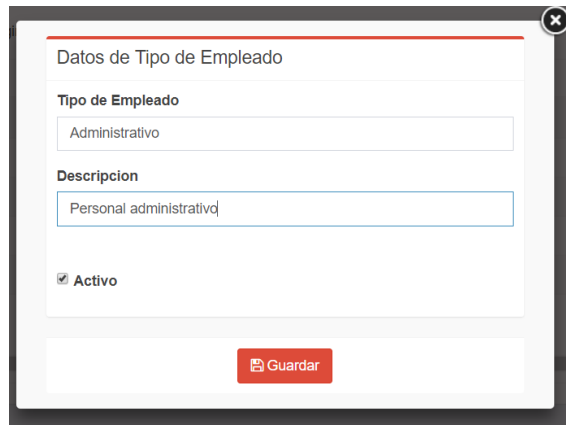
Mostrar 10 registros por página. Busqueda

Tipo de trabajador	Descripcion	Accion
Administrativo	Administrativo	 
Gerente	Gerente	 
Obrero	Obrero de producción	 
Pasante	Pasante	 
Practicante	Practicante	 

Mostrando página 1 de 1

Anterior 1 Siguiente

3. Clic sobre  en la línea correspondiente al tipo de trabajador que se desea editar.
4. Editar el tipo de trabajador o la descripción del mismo.



Datos de Tipo de Empleado

Tipo de Empleado  
Administrativo

Descripcion  
Personal administrativo

Activo

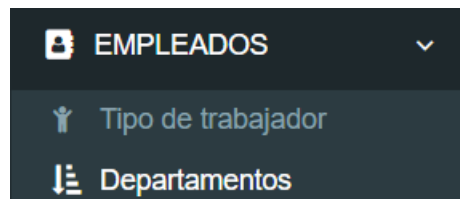
Guardar

5. Clic en

Guardar

#### 4.1.11. Crear departamento

1. Seleccionar la opción departamentos dentro del menú empleados



2. Clic sobre

Añadir

3. Escribir el nombre del departamento, y en el campo de jefe de departamento dejar vacío hasta crear un empleado que asuma dicho cargo

4. Clic en



#### 4.1.12. Editar departamento

1. Seleccionar la opción departamentos dentro del menú empleados



2. Se visualizará la lista de departamentos creados en el sistema.


Lista de Departamentos Inicio > Departamento

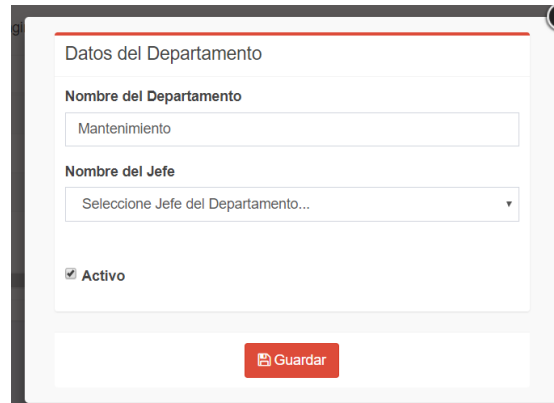
Datos de Departamentos + Añadir

Mostrar  registros por página. Búsqueda

Nombre de Departamento	Nombre de Jefe	Acción
Contabilidad		<input checked="" type="checkbox"/> <input type="checkbox"/>
Encalado		<input checked="" type="checkbox"/> <input type="checkbox"/>
Mantenimiento		<input checked="" type="checkbox"/> <input type="checkbox"/>
Sistemas		<input checked="" type="checkbox"/> <input type="checkbox"/>

Mostrando página 1 de 1 Anterior **1** Siguiente

3. Clic en el botón  en la línea correspondiente al departamento que se desea editar.
4. Editar el nombre del departamento y/o escoger el jefe departamental.



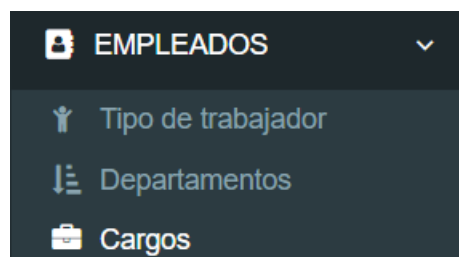
Formulario de edición de departamento con los siguientes campos:

- Título: Datos del Departamento
- Campo de texto: Nombre del Departamento (contiene 'Mantenimiento')
- Lista desplegable: Nombre del Jefe (contiene 'Seleccione Jefe del Departamento...')
- Checkbox: Activo (está marcado)
- Botón: Guardar

5. Clic en 

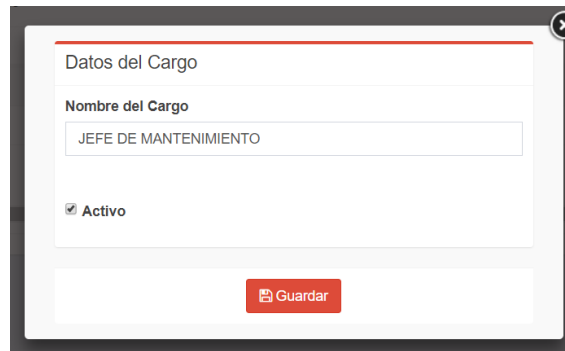
#### 4.1.13. Editar cargos

1. Seleccionar la opción cargos dentro del menú empleados



2. Clic sobre 

3. Escribir el nombre de cargo que se va a crear

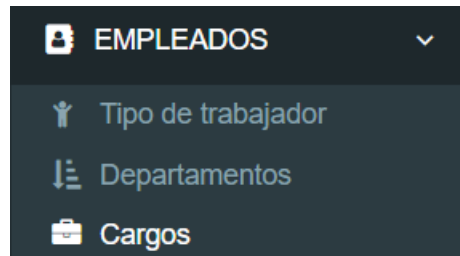


A screenshot of a web form titled "Datos del Cargo". The form has a white background and a thin red border. At the top right, there is a small circular close button with an 'x'. Below the title, there is a section labeled "Nombre del Cargo" with a text input field containing the text "JEFE DE MANTENIMIENTO". Below this, there is a checkbox labeled "Activo" which is checked. At the bottom center of the form, there is a red button with a white floppy disk icon and the text "Guardar".

4. Clic en 

#### 4.1.14. Modificar cargos

5. Seleccionar la opción cargos dentro del menú empleados



1. Se visualizará la lista de cargos creados en el sistema


Lista de Cargos Inicio > Cargos

Datos de Cargos + Añadir

Mostrar 10 registros por página. Búsqueda

Nombre de cargo	Accion
AUXILIAR CONTABLE	<input checked="" type="checkbox"/> <input type="checkbox"/>
JEFE DE MANTENIMIENTO	<input checked="" type="checkbox"/> <input type="checkbox"/>
JEFE DE SISTEMAS	<input checked="" type="checkbox"/> <input type="checkbox"/>
OPERADOR ACIDULADO	<input checked="" type="checkbox"/> <input type="checkbox"/>
OPERADOR DE MANTENIMIENTO	<input checked="" type="checkbox"/> <input type="checkbox"/>
OPERADOR ENCALADO	<input checked="" type="checkbox"/> <input type="checkbox"/>

Mostrando página 1 de 1 Anterior 1 Siguiente

2. Clic en el botón  en la línea correspondiente al departamento que se desea editar.

3. Editar el nombre del cargo

Datos del Cargo

Nombre del Cargo

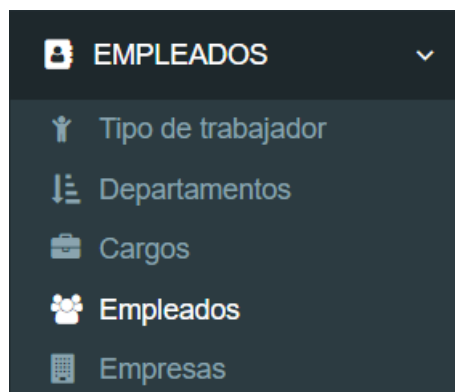
OPERADOR ACIDULADO


Activo

4. Clic en 

#### 4.1.15. Crear trabajador

1. Selecciona la opción empleados dentro del menú empleado



2. Clic sobre 
3. Se mostrará el formulario de datos del empleado, este cuenta con 4 secciones principales que son: Datos personales, datos generales, rubros y cargas familiares.

**Datos de Empleado** Inicio > Empleados

DATOS PERSONALES DATOS GENERALES RUBROS CARGAS FAMILIARES

Apellidos Apellidos	Nombres Nombres		Sexo Femenino	<input checked="" type="checkbox"/> Activo
Tipo Identificación Cédula	Identificación Identificación	Lugar Expedición Lugar Expedición		Pasaporte Pasaporte
Profesión Profesión	Calle Principal Calle Principal	Numero Vivienda Numero Vivienda	Calle Transversal Calle Transversal	Sector Sector
Tipo Vivienda Seleccione...	Ciudad Seleccione...	Telefono Telefono	Celular Celular	Correo Correo
Estado Civil Seleccione...	Fecha Nacimiento Fecha Nacimiento	Peso Peso	Talla Talla	Tipo Discapacidad Seleccione...
				Cedula Militar Cedula Militar
				Referencia Referencia
				Tipo Sangre Seleccione...
				%Discapacidad %Discapacidad

Vive con Familiares

#### 4. Llenar los datos personales del trabajador:

**Datos de Empleado** Inicio > Empleados

DATOS PERSONALES DATOS GENERALES RUBROS CARGAS FAMILIARES

Apellidos GONZALEZ PALACIOS	Nombres JORGE ESTEBAN		Sexo Mascullino	<input checked="" type="checkbox"/> Activo
Tipo Identificación Cédula	Identificación 1803071438	Lugar Expedición AMBATO		Pasaporte Pasaporte
Profesión ESTUDIANTE	Calle Principal AV RODRIGO PACHAN	Numero Vivienda S/N	Calle Transversal 22 DE ENERO	Sector LA VICTORIA
Tipo Vivienda Propia	Ciudad Ambato	Telefono 032842466	Celular 0992977489	Correo jgonzalez@ecuoyme.cc
Estado Civil Casado(a)	Fecha Nacimiento 24/08/1989	Peso 175 lb	Talla 165 cm	Tipo Discapacidad Seleccione...
				Cedula Militar Cedula Militar
				Referencia JUNTO A PANIFICADC
				Tipo Sangre A positivo
				%Discapacidad %Discapacidad

Vive con Familiares

#### 5. Llenar los datos de la persona de contacto del trabajador

Datos de Contacto

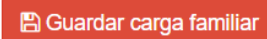
Nombres y Apellidos CRISTINA PINTO	Parentesco Conyuge	Direccion AV RODRIGO PACHANO Y 22 DE ENERO	Telefonos Telefonos
---------------------------------------	-----------------------	---	------------------------

6. En la pestaña de cargas familiares se debe agregar todas las cargas familiares del trabajador, para registrar una carga familiar pulsar sobre el

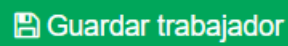
botón 

7. Llenar los datos de la carga familiar

8. Guardar la carga familiar pulsando sobre el botón

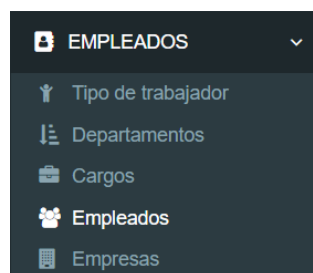


9. Guardar toda la información del trabajador pulsando sobre el botón



#### 4.1.16. Editar trabajador

1. seleccionar la opción empleados dentro del menú empleado









2. Se cargará una tabla con toda la información de los trabajadores registrados

**Lista de Empleados** Inicio > Empleados

Datos de Empleados Añadir trabajador

Mostrar 10 registros por página. Búsqueda

Apellidos	Nombres	Identificacion	Departamento	Telefono	Correo	Accion
GONZALEZ PALACIOS	JORGE ESTEBAN	1803071438001	Sistemas	0992977489	jorgecurlygp@gmail.com	 
PALACIOS GAVILANES	FAUSTO ANIBAL	1800033845		032841879	faustopalacios@yahoo.com	 
PALACIOS PEREZ	ROCIO MARIANELA	1801645027	Contabilidad	0993751832	rmarianelap@yahoo.com	 



Mostrando página 1 de 1 Anterior 1 Siguiente

3. Buscar el trabajador que desea editar, escribiendo su número de identificación o cualquier apellido o nombre en la barra de búsqueda


**Lista de Empleados** Inicio > Empleados


Datos de Empleados Añadir trabajador

Mostrar 10 registros por página. Búsqueda GONZALEZ

Apellidos	Nombres	Identificacion	Departamento	Telefono	Correo	Accion
GONZALEZ PALACIOS	JORGE ESTEBAN	1803071438001	Sistemas	0992977489	jorgecurlygp@gmail.com	 

Mostrando página 1 de 1 (filtrados de un total de 3 registros) Anterior 1 Siguiente

4. Pulsar sobre el botón  que se encuentra en el la columna acción del trabajador buscado
5. Editar cualquier campo del trabajador y guardar los cambios con el botón

 **Guardar trabajador**

#### 4.1.17. Asignar un contrato a un trabajador

1. Ingresar a editar el trabajador al que se necesita asignar un contrato
2. En la pestaña datos generales, en la sección de contrato, registrar un nuevo contrato de trabajo para el trabajador con el botón Añadir Contrato
3. Seleccionar fecha de ingreso, tipo de contrato, cargo asignado y salario

**Datos de Empleado** Inicio > Empleados

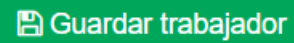
[DATOS PERSONALES](#)
[DATOS GENERALES](#)
[\\$ RUBROS](#)
[CARGAS FAMILIARES](#)

---

**Contrato**

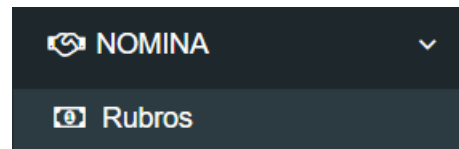
Fecha Ingreso	Tipo Contrato	Cargo	Sueldo
01/08/2018	Tiempo Indefinido	JEFE DE SISTEMAS	1200
<input type="checkbox"/> Fecha Salida	Causa de Salida		
Fecha Salida	Causa de Salida		

4. Guardar el contrato asignado con el botón

 Guardar trabajador

#### 4.1.18. Crear rubros

1. Seleccionar la opción rubros dentro del menú Nómina



2. Pulsar sobre

 Añadir Rubro

3. Escribir los datos del rubro, tales como nombre, código, tipo, periodicidad, mes activo, días de gracia, en caso de ser necesario usar una expresión de cálculo activar la casilla de calculado y escribir su fórmula

Datos de Rubro

Nombre de Rubro  
Sueldo Base  Activo

Codigo de Rubro: 0100 Tipo: Informativo

Periodicidad: Mensual Mes Activo: Todos

Dias de gracia: 0  Editable

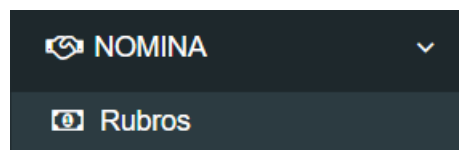
Calculado

Expresion de Calculo

4. Clic en

#### 4.1.19. Actualizar rubros

1. Seleccionar la opción rubros dentro del menú Nómina



















2. Se cargará la información de todos los rubros generados en el sistema


**Lista de Rubros** Inicio > Rubros

Datos de Rubros + Añadir Rubro

Mostrar 10 registros por página. Búsqueda

Codigo	Nombre	Tipo Rubro	Periodicidad	Calculado	Accion
0100	Sueldo Base	Informativo	Mensual	NO	 
0101	DIAS TRABAJADOS	Ingreso	Mensual	NO	 
1000	Sueldo devengado	Ingreso	Mensual	SI	 
1020	Valor Recargo Nocturno 25%	Informativo	Anual	SI	 
1030	Valor Horas Suplementarias 50%	Ingreso	Mensual	NO	 
3000	NETO A RECIBIR	Informativo	Mensual	SI	 
3010	APORTE PERSONAL IESS	Egreso	Mensual	SI	 
3020	APORTE PATRONAL IESS	Provisiones	Mensual	SI	 

Mostrando página 1 de 1 Anterior 1 Siguiente

- Clic sobre el botón  que está en la columna de acción en la línea correspondiente al rubro que se desea editar
- Editar los campos del rubro deseados

**Datos de Rubro**

Nombre de Rubro   Activo

Codigo de Rubro  Tipo

Periodicidad  Mes Activo

Dias de gracia

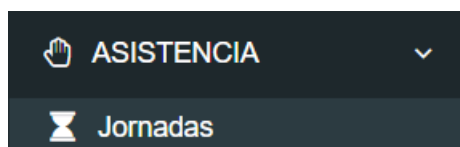
Calculado


Guardar Rubro

- Clic en Guardar Rubro

#### 4.1.20. Crear jornada de trabajo

- Clic sobre la opción jornadas dentro del menú asistencia



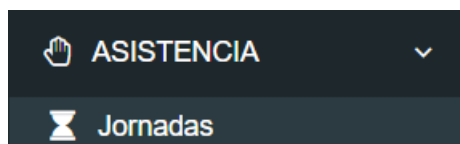
2. Clic sobre 
3. Llenar los datos de la jornada

A form titled 'Datos de jornada' with a close button in the top right corner. It contains a text input field for 'Descripcion de jornada' with the value 'ADMINISTRATIVO' and an 'Activo' checkbox. Below are six time input fields arranged in two columns: 'Entrada Empresa' (08:00), 'Salida Empresa' (16:30), 'Entrada Trabajo' (08:00), 'Salida Trabajo' (16:30), 'Salida Almuerzo' (12:30), and 'Entrada Almuerzo' (13:00). A red 'Guardar jornada' button is at the bottom.

4. Clic en 

#### 4.1.21. Actualizar jornada de trabajo

1. Clic sobre la opción jornadas dentro del menú asistencia





2. Se visualizará la lista de todas las jornadas laborables creadas


Lista de jornadas Inicio > Jornadas

Datos de Jornadas + Añadir Jornada

Mostrar 10 registros por página. Búsqueda

Descripcion	Entrada	Sal.Almuerzo	Ent.Almuerzo	Salida	Accion
ADMINISTRATIVO	08:00:00	13:00:00	14:00:00	16:00:00	 

Mostrando página 1 de 1 Anterior 1 Siguiente

- Clic sobre el botón  en la fila de la columna que se desea editar
- Actualizar el nombre o los horarios de la jornada

Datos de jornada

Descripcion de jornada  
  Activo

Entrada Trabajo  Salida Trabajo

Salida Almuerzo  Entrada Almuerzo

Guardar jornada

- Clic en Guardar jornada

#### 4.1.22. Asignar jornada de trabajo

- Editar trabajador al que se desea asignar la jornada laboral
- En la pestaña de datos generales, sección otros, seleccionar la jornada de trabajo

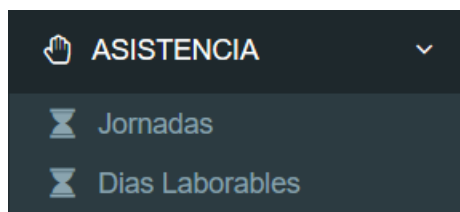
Otros

Perfil de Usuario <input type="text" value="Jefe Departamento"/>	Departamento <input type="text" value="Sistemas"/>	Empresa <input type="text" value="Seleccione..."/>	Jornada <input type="text" value="ADMINISTRATIVO"/> <input type="text" value="Seleccione..."/> <input style="background-color: #2980b9; color: white;" type="text" value="ADMINISTRATIVO"/>	Codigo Reloj <input type="text" value="Codigo Reloj"/>
---	---	---	--	---


3. Clic en 

#### 4.1.23. Configurar días laborables de cada mes

1. Seleccionar la opción Días Laborables del menú Asistencia



2. Seleccionar el año deseado, y editar los días laborables en cada uno de los meses.

 Registro de Días Laborables

Año

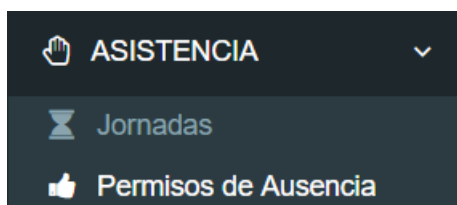
Mostrar  registros por página.

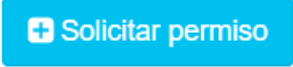
#	Mes	Días Laborables
1	Enero	<input type="text" value="31"/>
2	Febrero	<input type="text" value="28"/>
3	Marzo	<input type="text" value="31"/>
4	Abril	<input type="text" value="30"/>
5	Mayo	<input type="text" value="31"/>
6	Junio	<input type="text" value="30"/>
7	Julio	<input type="text" value="31"/>
8	Agosto	<input type="text" value="31"/>
9	Septiembre	<input type="text" value="30"/>
10	Octubre	<input type="text" value="31"/>

Mostrando página 1 de 2

#### 4.1.24. Solicitar permiso de ausencia

1. Ingresar en la opción permisos de ausencia del menú asistencia



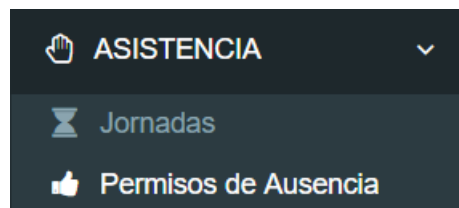
2. Clic en 
3. El sistema identificara el usuario con el que se ha iniciado sesión, se deberá seleccionar la fecha y hora de salida, así como la fecha y hora de regreso y el motivo del permiso

4. Clic en 

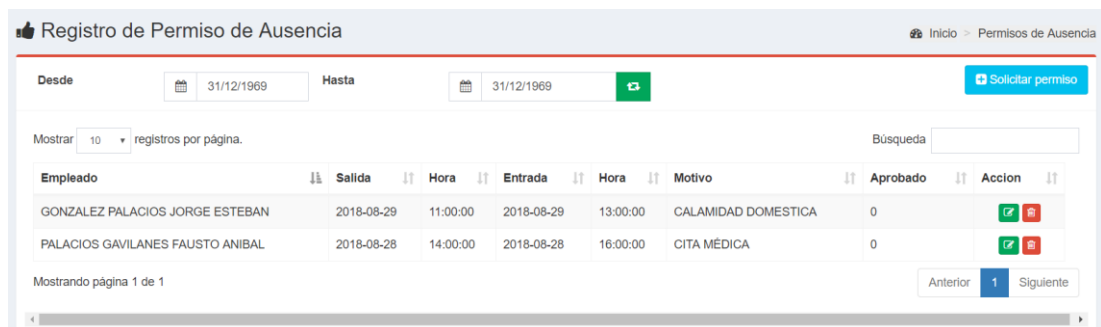
#### 4.1.25. Aprobar permisos de ausencia


Esta opción aparecerá en los usuarios que son jefes departamentales

1. Seleccionar la opción permisos de ausencia dentro del menú asistencia



2. El sistema listará todos los permisos de los usuarios que pertenezcan al departamento del cual el usuario es jefe



3. Pulsar sobre el botón  en la fila del permiso que se desea aprobar

4. Activar la casilla de aprobado

Datos de Permiso de Ausencia

Empleado: GONZALEZ PALACIOS JORGE ESTEBAN  Aprobado

Fecha Salida: 29/08/2018 Hora Salida: 11:00

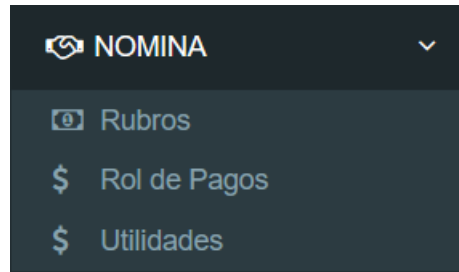
Fecha Entrada: 29/08/2018 Hora Entrada: 13:00


Motivo: CALAMIDAD DOMESTICA

5. Clic en 

#### 4.1.26. Creación de rol de pagos

1. Ingresar en la opción rol de pagos dentro del menú nómina



2. Clic en  Añadir rol
3. Seleccionar fecha de inicio y fecha de fin de rol de pagos.
4. Seleccionar fecha de inicio y fin de asistencia del personal
5. Escribir una descripción del rol de pagos
6. El sistema enlistará todos los empleados registrados en la nómina, así como los rubros asignados a los mismos y calculará el valor a recibir

**Datos de Rol de Pagos** Inicio > Roles de Pago

Inicio Rol: 01/08/2018    Fin Rol: 31/08/2018    Inicio Asistenc.: 05/07/2018    Fin Asistencia: 05/07/2018    Guardar

Descripción: ROL DE PAGOS AGOSTO 2018

Empleados				
#	Apellidos	Nombres	Monto	Ver
1	GONZALEZ PALACIOS	JORGE ESTEBAN	1610.40	
2	PALACIOS PEREZ	ROCIO MARIANELA	753.60	

Rubros			
Mostrar	10	registros por página.	Búsqueda <input type="text"/>
Codigo	Nombre	Valor	
Lo sentimos. No se encontraron registros.			
No hay registros aún.			
		Anterior	Siguiente

7. Para ver el detalle de rubros de un trabajador se deberá dar clic en el empleado en la tabla empleados, y el sistema enlistará los valores y sus descripciones en la tabla rubros

Empleados					Rubros		
#	Apellidos	Nombres	Monto	Ver	Mostrar	registros por página.	Búsqueda
1	GONZALEZ PALACIOS	JORGE ESTEBAN	1610.40		Codigo	Nombre	Valor
2	PALACIOS PEREZ	ROCIO MARIANELA	633.60		0100	Sueldo Base	800.00
					0101	DIAS TRABAJADOS	20.00
					1000	Sueldo devengado	800.00
					3000	NETO A RECIBIR	633.60
					3010	APORTE PERSONAL IESS	76.80
					3020	APORTE PATRONAL IESS	89.60

Mostrando página 1 de 1      Anterior 1 Siguiete

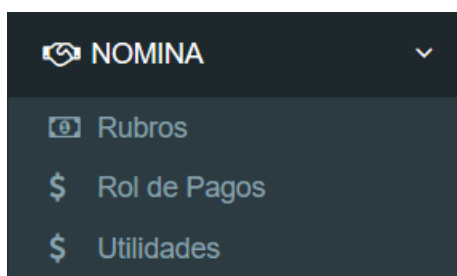
8. Para generar la representación impresa de rol de pagos, dar clic sobre el



botón

Prodegeu		PLANILLA INDIVIDUAL AL 2018-08-31		Fecha: 2018-08-27	
		PALACIOS PEREZ ROCIO MARIANELA			
		AUXILIAR CONTABLE			
<b>DIAS TRABAJADOS: 20</b>					
<b>INGRESOS:</b>					
Sueldo devengado		800.00	<b>DESCUENTOS:</b>		
Valor Horas Suplementarias 50%		0	APORTE PERSONAL IESS		76.80
<b>TOTAL INGRESOS:</b>		<b>800.00</b>	<b>TOTAL DESCUENTOS:</b>		<b>76.80</b>
<b>Neto a Recibir: 633.60</b>					

#### 4.1.27. Impresión de rol de pagos masivo

1. Ingresar en la opción Rol de pagos dentro del menú Nómina

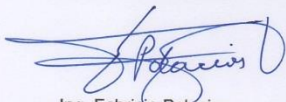


3. Dar clic en  en la fila del rol que se desea imprimir masivamente para todos los trabajadores.
4. Clic en 
5. El sistema generará un archivo en formato pdf, que contendrá todos los roles de pagos de los trabajadores.

## 4.2. Pruebas Integrales

Productora de Gelatina Ecuatoriana S.A.  
RUC. 1792183154001



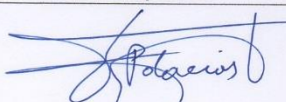
Módulo de configuración		
Encargado de la prueba	Definición de prueba	Resultados
Gerente General	<ol style="list-style-type: none"> <li>1. Usuarios <ul style="list-style-type: none"> <li>• Crear usuario</li> <li>• Actualizar usuario</li> </ul> </li> <li>2. Países <ul style="list-style-type: none"> <li>• Crear países</li> <li>• Actualizar países</li> </ul> </li> <li>3. Provincias <ul style="list-style-type: none"> <li>• Crear provincias</li> <li>• Editar provincias</li> </ul> </li> <li>4. Ciudades <ul style="list-style-type: none"> <li>• Crear ciudades</li> <li>• Editar ciudades</li> </ul> </li> <li>5. Tipos de sangre <ul style="list-style-type: none"> <li>• Crear tipos de sangre</li> <li>• Editar tipos de sangre</li> </ul> </li> <li>6. Bancos <ul style="list-style-type: none"> <li>• Crear bancos</li> <li>• Editar bancos</li> </ul> </li> <li>7. Parámetros <ul style="list-style-type: none"> <li>• Configurar parámetros del sistema</li> </ul> </li> </ol>	<p>Para cada elemento del módulo de configuración, se realizan acciones de inserción y consultas de datos, se testea que los parámetros aplicados en las configuraciones surtan los efectos deseados.</p> <p>Se ingresa al sistema con todos los usuarios creados.</p>
<b>Firma de aceptación</b>		
 Ing. Fabricio Palacios Gerente General PRODEGEL S.A.		



Kilómetro 10 1/2 Vía a Baños / Talara / Cantón Pelileo / Ambato - Ecuador  
Teléfonos: 03 - 2748118 / 03 - 2748123 / 03 - 2748124 / Fax: 03 - 2748122

Productora de Gelatina Ecuatoriana S.A.  
RUC. 1792183154001




Módulo de empleados		
Encargado de la prueba	Definición de prueba	Resultados
Gerente General	<ol style="list-style-type: none"> <li>1. Tipo de trabajador <ul style="list-style-type: none"> <li>• Crear tipo de trabajador</li> <li>• Actualizar tipo de trabajador</li> </ul> </li> <li>2. Departamentos <ul style="list-style-type: none"> <li>• Crear departamentos</li> <li>• Actualizar departamentos</li> </ul> </li> <li>3. Cargos <ul style="list-style-type: none"> <li>• Crear cargos</li> <li>• Editar cargos</li> </ul> </li> <li>4. Empresas <ul style="list-style-type: none"> <li>• Crear empresas</li> <li>• Editar empresas</li> </ul> </li> <li>5. Empleados <ul style="list-style-type: none"> <li>• Crear empleados</li> <li>• Editar empleados</li> </ul> </li> </ol>	Para cada elemento del módulo de empleados, se realizan acciones de inserción y consultas de datos, y se asignan al trabajador todos los datos creados en la prueba anterior.
<b>Firma de aceptación</b>		
 Ing. Fabricio Palacios Gerente General PRODEGEL S.A.		



Kilómetro 10 1/2 Vía a Baños / Totoras / Cantón Peñileo / Ambato - Ecuador  
Teléfonos: 03 - 2748118 / 03 - 2748123 / 03 - 2748124 / Fax: 03 - 2748122

Productora de Gelatina Ecuatoriana S.A.  
RUC. 1792183154001



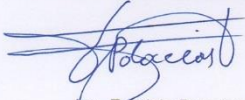
Módulo de asistencia		
Encargado de la prueba	Definición de prueba	Resultados
Gerente General	<ol style="list-style-type: none"> <li>1. Jornadas <ul style="list-style-type: none"> <li>• Crear jornadas</li> <li>• Actualizar jornadas</li> <li>• Asignar jornadas a trabajador</li> </ul> </li> <li>2. Permisos de ausencia <ul style="list-style-type: none"> <li>• Crear tipos de permiso de ausencia</li> <li>• Editar tipo de permiso de ausencia</li> <li>• Solicitar permiso ausencia</li> <li>• Autorizar permiso de ausencia</li> <li>• Importar registros de ausencia</li> </ul> </li> </ol>	Para cada elemento del módulo de asistencia, se realizan acciones de inserción y consultas de datos.
<b>Firma de aceptación</b>		
 Ing. Fabricio Palacios Gerente General PRODEGEL S.A.		



Kilómetro 10 1/2 Vía a Baños / Tatoras / Cantón Pelileo / Ambato - Ecuador  
Teléfonos: 03 - 2748118 / 03 - 2748123 / 03 - 2748124 / Fax: 03 - 2748122

Productora de Gelatina Ecuatoriana S.A.  
RUC. 1792183154001



Módulo de nómina		
Encargado de la prueba	Definición de prueba	Resultados
Gerente General	1. Rubros <ul style="list-style-type: none"> <li>• Crear rubros</li> <li>• Actualizar rubros</li> <li>• Asignar rubros a trabajador</li> </ul> 2. Roles de pago <ul style="list-style-type: none"> <li>• Crear rol de pagos</li> <li>• Editar rol de pagos</li> <li>• Imprimir rol de pagos</li> </ul>	Para cada elemento del módulo de nómina, se realizan acciones de inserción y consultas de datos.
<b>Firma de aceptación</b>		
 Ing. Fabricio Palacios Gerente General PRODEGEL S.A.		



Kilómetro 10 1/2 Vía a Baños / Tojoras / Cantón Pelileo / Ambato - Ecuador  
Teléfonos: 03 - 2748118 / 03 - 2748123 / 03 - 2748124 / Fax: 03 - 2748122

Productora de Gelatina Ecuatoriana S.A.  
RUC. 1792183154001



Ambato, 30 de agosto de 2018

Yo, DIEGO FABRICIO PALACIOS PEREZ, en calidad de GERENTE GENERAL de la EMPRESA PRODUCTORA DE GELATINA ECUATORIANA S.A. PRODEGEL S.A., certifico que el Sr. JORGE ESTEBAN GONZÁLEZ PALACIOS, portador de la cédula de ciudadanía # 180307143-8, ha desarrollado el sistema PRODEGELRRHH, para aplicación en los procesos de: Gestión de Personal, Gestión de Nómina y Asistencia, en el Área de Talento Humano, luego de las pruebas integrales de cada módulo, se aprueba y recibe la herramienta con absoluta satisfacción.

Atentamente,

A handwritten signature in blue ink, appearing to read "Fabricio Palacios Pérez", is written over a horizontal line.

Ing. Fabricio Palacios Pérez

GERENTE GENERAL - PRODEGEL S.A.



Kilómetro 10 1/2 Vía a Baños / Tatoras / Cantón Pelileo / Ambato - Ecuador  
Teléfonos: 03 - 2748118 / 03 - 2748123 / 03 - 2748124 / Fax: 03 - 2748122

## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

Después del desarrollo e implementación del *software*, se concluye que:

La metodología FDD, es una herramienta útil para el desarrollo de *software* de forma colaborativa, que permite la interacción efectiva entre un grupo amplio de personas con distintos roles y actividades a ejecutar, en el proyecto.

La herramienta implementada, será útil para la gestión de las actividades diarias del Área de Talento Humano, aportará para la ejecución efectiva de los procesos con una reducción importante en la inversión de tiempo del personal.

La información centralizada, actualizada y oportuna, que este *software* brindará, va a ser un instrumento de alto valor, para el giro del negocio de Prodegel, pues le permitirá ser competitivo al poder tomar decisiones oportunas, en relación a sus colaboradores.

Se recomienda:

Registrar la información oportunamente, a fin de que se aproveche la herramienta que se está suministrando a Prodegel.

El recurso humano, es el valor principal en todas las empresas, por lo que es importante buscar alternativas para alcanzar un buen nivel de clima laboral y un puntal importante es la información otorgada a través del rol de pagos, por tanto, se deben generar todos los rubros requeridos a fin de lograr el nivel de detalle de la información que permita ser entendido por cualquier trabajador.

## BIBLIOGRAFÍA

- Anil, J., Arun, R., & Flinn, P. (2008). Libro de bolsillo de biométricos. New York: Springer.
- Blé, C. (2010). *Diseño Ágil con TDD: Test Driven Developmnet*. Bogotá: www.iExpertos.com.
- Chiavenato, I. (2002). *Gestión del Talento Humano*. México: McGraw-Hill
- Cobo, A., Gómez, P., Pérez, D., & Rocha R. (2005). *PHP y MYSQL Tecnologías para el desarrollo de aplicaciones web*. Díaz de Santos.
- Coronel. C., Morris S., & Rob, P. (2011). *Bases de Datos, Diseño, Implementación y Administración (9ª ed.)*. México, D.F.: Cengage Learning Editores S.A.
- Date, C. (2001). *Sistemas de bases de datos*. México D.F.: Pearson Prentice Hall.
- Douglas, K. (2003). *PostgreSQL una comprensiva guía para construir, programar, y administrar bases de datos PostgreSQL*. Indianápolis: Sams Publishing.
- Chrome. (s.f.) *Draw.io*. <https://www.draw.io/>
- Ellislab. (s.f.). *CodeIgniter*. <https://ellislab.com/>
- Epordo. (2013). *Manual de Usuario para la serie ET*. China: E-Focus Industrial Co., Ltd.
- Eslava Muñoz, V. (2016). *El nuevo PHP. Conceptos avanzados*. Madrid: Bubok Publishing S.L.
- Falgueras, B. C. (2003). *Ingeniería de software*. Barcelona: Editorial UOC.
- Flores, J. J. (2005). *Método de las 6'D. UML – Pseudocódigo – Java. (Enfoque algorítmico)*. Perú: Editorial - Universidad San Martín de Porres.
- Forum CodeIgniter (s.f.). *Forums*. <https://forum.codeigniter.com/>
- Foundation, I. E. (s.f.). *Eclipse - The Eclipse Foundation open source community website*. <http://www.eclipse.org/>
- Fowler, M., Scott, K. (1999). *UML gota a gota*. Bogotá: Pearson Education.
- Gómez, J. (2016). *Dirección y gestión de proyectos de tecnologías de la información en la empresa*. Madrid: Fundación Confemetal.

- González, R. (2005). *Nuevas tecnologías aplicadas a la gestión de RRHH: las TIC's como herramienta de mejora permanente del capital humano*. Vigo: IdeasPropias.
- Goñi, J. J. (2012). *Talento, tecnología y tiempo*. Madrid: Ediciones Díaz de Santos.
- Goyal, S. (2007). *Agile Techniques for Project Management and Software Engineering*. Tesis de Ingeniería de Software. Technical University Munich.
- Gracia, E. (2016). *Aprende a Modelar Aplicaciones con UML: 2º Edición*. Madrid: IT Campus Academy.
- Granados, R. L. (2014). *Despliegue y puesta en funcionamiento de componentes software*. Málaga: IC Editorial.
- GanttProject. (s.f.). Recuperado 09 de Julio de 2017, de <http://www.ganttproject.biz/about>.
- Jain, A. K., Flynn, F. & Ross, A. A. (2008). *Handbook of Biometrics*. New York: Springer.
- Lainez, J. R. (2014). *Desarrollo de Software Ágil: Extreme Programming y Scrum*. Madrid: ITL Campus Academy.
- Letelier, P. & Sánchez, E. (2003). *Metodologías Ágiles en el Desarrollo de Software*. Alicante: Grupo ISSI.
- Leyva, E., Prieto, J. I., Sampalo, M. & Garzón, M. L. (2006). *Sistemas y Aplicaciones Informáticas*. Madrid: Editorial MAD S.L.
- Martin, S. A. (2011). *PostgreSQL: Una ponderosa base de datos libre*. Estados Unidos: EEA Editorial Academia Española.
- Martínez, R., Laínez, J. R., Durango, A. & Ramos, D. (2015). *Curso de Ingeniería de Software*. Madrid: ITL Campus Academy.
- Martínez, R., Laínez, J. R., Durango, A. & Ramos, D. (2017). *Curso de Ingeniería de Software: 2ª Edición*. Madrid: ITL Campus Academy.
- Metodología FDD. (s.f.). *Metodología FDD*. (s.f.). Recuperado 17 de enero de 2017, de <http://metodologiafdd.blogspot.com/>
- MICROSOFT. (s.f.). *Diagrama de Secuencia UML: Referencia*. Recuperado 5 de julio de 2017, de <https://msdn.microsoft.com/es-es/library/dd409377.aspx>.

- Montes, M. J. & González, P. G. (2006). *Selección de Personal: La búsqueda del candidato adecuado*. España: Editorial Ideas Propias
- Mora, S. L. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante: Editorial Club Universitario.
- Moro, M. & Rodés, A. (2014). *Marketing digital: Comercio y marketing*. España: Ediciones Paraninfo S.A.
- NEBULON. (s. f.). *Feature Driven Development (FDD)*.  
<http://www.nebulon.com/fdd/index.html>
- NEBULON. (s. f.). *Feature Driven Development (FDD)*.  
<http://www.featuredrivendevelopment.com/>
- Palmer, S. & Felsing, J. (2002). *A Practical Guide to Feature-driven Development*. Upper Saddle River, NJ: Prentice Hall PTR
- PostgreSQL. (s. f.). *The world's most advanced open source database*.  
<https://www.postgresql.org/about/>.
- Pressman, R. S., Olguín, V. C., & Brito, J. E. (2010). *Ingeniería del software: un enfoque práctico* (7ma ed.). México: McGraw-Hill.
- Quora. (s. f.). *What are the differences between HTML, XML, PHP, CSS and JavaScript in layman terms?* Recuperado 18 de enero de 2017, de <https://www.quora.com/What-are-the-differences-between-HTML-XML-PHP-CSS-and-JavaScript-in-layman-terms>
- Rodríguez, J. J.. (2003). *Introducción a la programación: teoría y práctica*. San Vicente (Alicante): Club Universitario.
- Schultz, R. (2009). *Diseño web con CSS*. Barcelona: MARCOMBO S.A.

## **GLOSARIO**

### **BSD**

La licencia BSD (*Barkley Software Distribution*), creada por la Universidad de Barkley, es una de las menos restrictivas. Permite la redistribución y distribución del *software* y no impone restricciones a cómo puede ser utilizado el código.

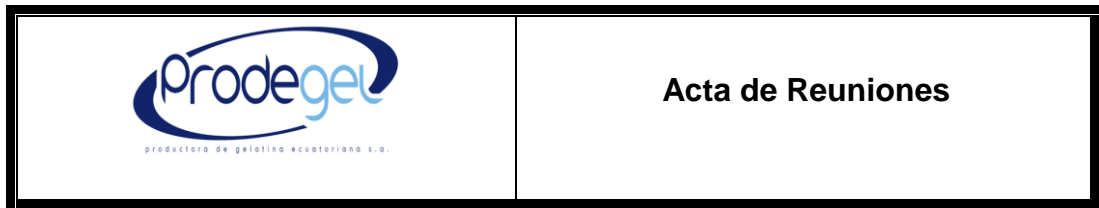
### **UML (Unified Modeling Language)**

Lenguaje para modelar objetos, es un lenguaje gráfico para representar, explicar y documentar cada una de las partes que comprende el desarrollo de *software*.

## ANEXOS

### Anexo 1: Documentos Adicionales – Walkthrough General

#### 1. ACTA DE REUNIÓN N. 001



<b>Motivo:</b> Levantar información para elaboración de walkthrough	<b>Acta N. 001</b>	<b><u>Versión: 1</u></b>
Lugar de reunión: Prodegel		
Fecha de reunión: 29.05.2017		
Hora de reunión: 09:00 – 13:00		

**Funcionarios fijos a la reunión: P = PRESENTE    A = AUSENTE**

<b>Nombre</b>	<b>Cargo</b>	<b>Rol</b>	
Fabricio Palacios	Gerente General	Sponsor del Proyecto	P
Pamela Morales	Analista de Recursos Humanos	Experto de Dominio	P
Jorge González		Administrador del Proyecto	P

**ORDEN DEL DIA:**

1.- Presentación de la Empresa
2.- Presentación del Área de Talento Humano
3.- Levantamiento de problemas

**PUNTOS TRATADOS:**

## 1. Presentación de la Empresa

El Gerente General autoriza la entrega de documentación y archivos, en relación a: Misión, Visión, Objetivos y Organigramas.

## 2. Presentación del Área de Talento Humano

Se realiza la presentación de toda el Área de Talento Humano, con una explicación de las funciones que cumple cada persona.

## 3. Levantamiento de Problemas

Con la presencia de todo el personal del Área de talento Humano y el Gerente General, se enlistan los problemas, para luego priorizarlos en la Matriz Vester y finalmente generar un árbol de problemas.

**CONCLUSIONES ADICIONALES:**

**RESOLUCIONES PENDIENTES PARA LA SIGUIENTE REUNION**

**P = PENDIENTE**

**C = CONCLUIDO**

<b>#</b>	<b>RESOLUCION</b>	<b>RESPONSABLE</b>	<b>P</b>	<b>C</b>	<b>FECHA</b>
1	Elaborar los walktrough	Jorge González	X		02.06.2017

#### 4. MATRIZ VESTER

El Área de Talento Humano, tiene varios problemas que requieren resolver, en relación a los procesos que el área ejecuta en el día a día, pues los mismos afectan al personal en general de Prodegel, así como a la misma Empresa en relación a las obligaciones que debe cumplir con los entes gubernamentales.

##### Lista de Problemas

Código	Problema
P1	Información desactualizada del personal activo
P2	Descontento de los colaboradores con el rol de pagos impreso, porque no tiene el detalle completo de ingresos y egresos
P3	Gran inversión de tiempo para generar reportes de asistencia
P4	Falta de reportes automatizados para toma de decisiones
P5	Demasiado tiempo asignado a registrar permisos del personal
P6	Mucho tiempo entregado a actividades manuales para el cálculo de utilidades
P7	El rol de pagos tarda demasiado tiempo en ser calculado para el pago mensual
P8	<i>Software</i> antiguo que no permite mejoras
P9	No se puede aplicar con facilidad los constantes cambios de la normativa gubernamental
P10	Falta de soporte oportuno al <i>software</i> existente
P11	Problemas para controlar las vacaciones pendientes del personal

**Tabla:** 1. Lista de problemas  
**Elaborado por:** Jorge González

## Ficha Técnica

Detalle	Ficha Técnica
Código	P1
Enunciado del Problema	Información desactualizada del personal activo
Descripción	La información de los empleados, sufre cambios constantes, sin embargo, debido a que el personal del Área está ocupado en otras tareas, no se actualiza de forma constante
Tendencia	Incremento de los empleados con información desactualizada
Fuente de datos	Entrevista realizada al personal del Área de Talento Humano
Código	P2
Enunciado del Problema	Descontento de los colaboradores con el rol de pagos impreso, porque no tiene información detallada
Descripción	El <i>software</i> actual no permite realizar cambios al diseño del rol de pagos impreso, lo que hace que este no brinde el detalle de información requerido por los colaboradores
Tendencia	A incrementar el descontento de los colaboradores
Fuente de datos	Quejas de los colaboradores y continuas consultas al personal del Área de Talento Humano
Código	P3
Enunciado del Problema	Gran inversión de tiempo para generar reportes de asistencia
Descripción	Mensualmente para obtener el reporte de asistencia para control y pago de horas extras, es necesario dar tratamiento a los datos de los relojes con la ayuda de Excel.
Tendencia	Mantener el tiempo dedicada a dar formato al reporte de asistencia

Fuente de datos	Entrevista realizada al personal del Área de Talento Humano
Código	P4
Enunciado del Problema	Falta de reportes automatizados para toma de decisiones
Descripción	Gran cantidad de reportes que requiere la Gerencia, se deben generar previamente en Excel
Tendencia	Mantener el tiempo dedicado a generar reportes
Fuente de datos	Entrevista realizada al personal del Área de Talento Humano
Código	P5
Enunciado del Problema	Demasiado tiempo asignado a registrar permisos del personal
Descripción	El <i>software</i> actual no tiene control de permisos, por lo que estos se deben registrar en Excel, lo que no permite un control rápido y descuento de los días al instante de la nómina
Tendencia	Incremento en el tiempo dedicado a esta tarea
Fuente de datos	Entrevista realizada al personal del Área de Talento Humano
Código	P6
Enunciado del Problema	Mucho tiempo entregado a actividades manuales para el cálculo de utilidades
Descripción	Todos los años previo al pago de utilidades se debe actualizar los datos de cargas, días laborables, etc de todo el personal, en Excel y luego proceder con el cálculo
Tendencia	Mantener el tiempo entregado a esta actividad
Fuente de datos	Entrevista realizada al personal del Área de Talento Humano
Código	P7

Enunciado del Problema	El rol de pagos tarda demasiado tiempo en ser calculado para el pago mensual
Descripción	Para la generación mensual del rol de pagos, es necesario realizar cálculos de los diferentes rubros por fuera del <i>software</i> , para luego cargarlos.
Tendencia	Incremento en el tiempo requerido para elaboración del rol de pagos
Fuente de datos	Entrevista realizada al personal del Área de Talento Humano
Código	P8
Enunciado del Problema	<i>Software</i> antiguo que no permite mejoras
Descripción	No es posible incrementar módulos para apoyo a los procesos del Área de Talento Humano
Tendencia	A mantener igual el apoyo que brinda el <i>software</i> al Área
Fuente de datos	Entrevista realizada al personal del Área de Talento Humano
Código	P9
Enunciado del Problema	No se puede aplicar con facilidad los constantes cambios de la normativa gubernamental
Descripción	Cuando existen cambios es necesario buscar alternativas para solventarlos, debido a que el <i>software</i> no permite cambios a la configuración con facilidad
Tendencia	A mantenerse igual
Fuente de datos	Entrevista realizada al personal del Área de Talento Humano
Código	P10
Enunciado del Problema	Falta de soporte oportuno, al <i>software</i> existente

Descripción	Existen problemas con el <i>software</i> , que no se solventan a tiempo y paralizan los procesos
Tendencia	A mantenerse igual
Fuente de datos	Entrevista realizada al personal del Área de Talento Humano
Código	P11
Enunciado del Problema	Problemas para controlar las vacaciones pendientes del personal
Descripción	El <i>software</i> actual, no permite registrar los días de vacación tomadas por el empleado, lo que hace difícil el control
Tendencia	A incrementarse a medida que los colaboradores tienen derecho a vacaciones
Fuente de datos	Entrevista realizada al personal del Área de Talento Humano

**Tabla: 2.** Ficha Técnica  
**Elaborado por:** Jorge González

## Matriz

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	X
P1	0	0	0	3	0	3	1	3	0	0	0	10
P2	0	0	1	0	1	0	3	3	0	0	0	8
P3	0	2	0	0	3	0	3	3	0	0	0	11
P4	3	1	2	0	0	1	1	3	1	0	1	13
P5	0	0	3	0	0	0	3	3	0	0	0	9
P6	3	0	0	1	0	0	0	3	1	0	0	8
P7	1	3	3	0	3	0	0	3	2	3	0	18
P8	1	3	3	3	3	3	3	0	3	0	3	25

<b>P9</b>	2	3	1	1	1	1	3	3	0	2	0	17
<b>P10</b>	2	3	0	3	0	0	3	0	0	0	0	11
<b>P11</b>	0	0	0	0	0	0	0	3	0	0	0	3
<b>Y</b>	12	15	13	11	11	8	20	27	7	5	4	

**Tabla:** 3. Matriz Vester  
**Elaborado por:** Jorge González

Influencia = X

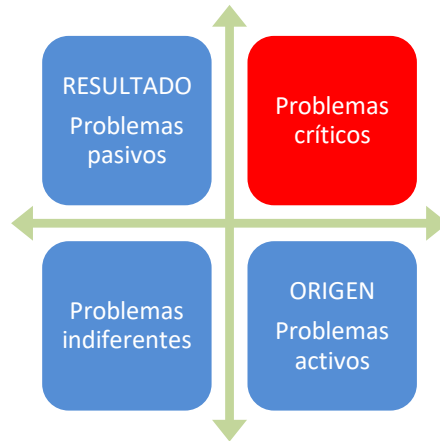
Dependencia = Y

### Plano Cartesiano

Problema	X	Y
<b>P1</b>	10	12
<b>P2</b>	8	15
<b>P3</b>	11	13
<b>P4</b>	13	11
<b>P5</b>	9	11

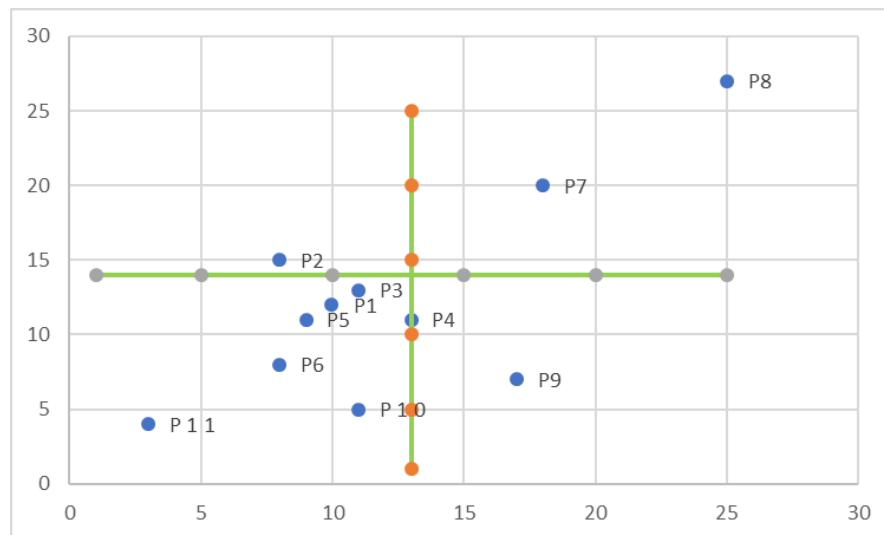
<b>P6</b>	8	8
<b>P7</b>	18	20
<b>P8</b>	25	27
<b>P9</b>	17	7
<b>P10</b>	11	5
<b>P11</b>	3	4

Clasificación de problemas: P8 es el problema con el número más alto en X, por tanto, se divide para 2, el resultado es  $(25/2=12.5)$  que equivale a 13, por lo que se traza una recta paralela al eje X, en ese punto. Se hace lo mismo para el eje Y, con P8 que es el más alto,  $(28/2=14)$  y se traza la línea paralela a Y.



**Figura: 1.11** Matriz de problemas

**Fuente:** Jorge González



**Figura: 2.12** Plano Cartesiano

**Elaborado por:** Jorge González

## Clasificación de Problemas

Código	Problema	Problema Central	Causas	Efectos
P1	Información desactualizada del personal activo			
P2	Descontento de los colaboradores con el rol de pagos impreso, porque no tiene información detallada			P2
P3	Gran inversión de tiempo para generar reportes de asistencia			
P4	Falta de reportes automatizados para toma de decisiones			
P5	Demasiado tiempo asignado a registrar permisos del personal			
P6	Mucho tiempo entregado a actividades manuales para el cálculo de utilidades			
P7	El rol de pagos tarda demasiado tiempo en ser calculado para el pago mensual		P7	
P8	<i>Software</i> antiguo que no permite mejoras	P8		
P9	No se puede aplicar con facilidad los constantes cambios de la normativa gubernamental		P9	
P10	Falta de soporte oportuno, al <i>software</i> existente			
P11	Problemas para controlar las vacaciones pendientes del personal			

**Tabla:** 4. Clasificación de Problemas

Elaborado por: Jorge González

## Anexo 2: Documentos Adicionales – Walkthrough por Módulos

### 1. ACTA DE REUNIÓN 002



Motivo: Levantar información para elaboración de walkthrough	Acta N. 002	<a href="#">Versión: 1</a>
Lugar de reunión: Prodegel		
Fecha de reunión: 02.06.2017		
Hora de reunión: 08:00 – 16:00		

**Funcionarios fijos a la reunión: P = PRESENTE    A = AUSENTE**

Nombre	Cargo	Rol	
Fabricio Palacios	Gerente General	Sponsor del Proyecto	P
Pamela Morales	Analista de Recursos Humanos	Experto de Dominio	P
Jorge González		Administrador del Proyecto	P

## **ORDEN DEL DIA:**

<b>1.- Módulo Empleados</b>
<b>2.- Módulo Nómina</b>
<b>3.- Módulo Asistencia</b>

## **PUNTOS TRATADOS:**

### **1. Módulo Empleados**

En esta sección se realizará el registro de los empleados de Prodegel, así como de los empleados de las otras Empresas complementarias, que tienen derecho a percibir utilidades.

La información que se requiere registrar de los empleados es:

## **PERSONAS**

### **Identificación General**

- Código
- Apellidos
- Nombres
- Identificación
- Lugar de Expedición
- Empresa
- Cédula Militar
- Pasaporte
- Correo Electrónico

### **Información Personal**

- Fecha de Nacimiento
- Profesión
- Nivel de Educación

- Sexo
- Estado Civil
- Tipo de Sangre
- Peso
- Talla

#### Teléfonos

- Domicilio
- Celular

#### Residencia

- Provincia
- Región
- Calle Principal
- Número
- Transversal
- Sector
- Referencia
- Tipo de Vivienda
- Vive con Familiares

#### Discapacidad

- Tipo
- Porcentaje

#### Cuenta Bancaria

- Banco
- Tipo
- Número

#### Persona de Contacto

- Nombre
- Dirección
- Parentesco
- Teléfono

#### Cargas Familiares

- Nombre del Familiar
- Identificación
- Parentesco
- Fecha de nacimiento
- Fecha de fallecimiento
- Fecha matrimonio
- Sexo
- Teléfono

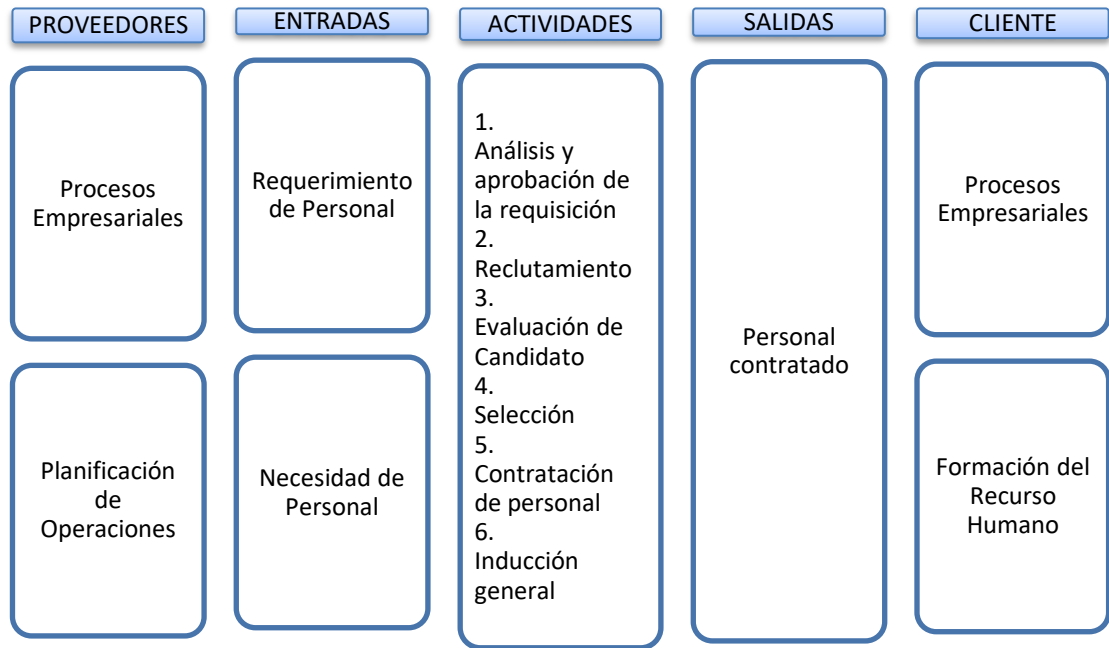
## CONTRATOS

- Código
- Fecha Ingreso
- Fecha Salida
- Sueldo
- Empresa
- Tipo de Contrato
- Área
- Departamento
- Cargo
- Tipo de Trabajo
- Motivo de Salida
- Causa de Salida

Se requieren, reportes de:

- Personal Activo, por Área, por Empresa, por Tipo de Contrato
- Cargas Familiares

## Caracterización del proceso de Incorporación del Recurso Humano



**Tabla:** 5. Proceso de Incorporación de Recurso Humano  
**Elaborado por:** Jorge González

## 2. Módulo Nómina

Se debe generar una nómina por mes, con todos los empleados activos en el período, que contengan los rubros de ingresos, egresos y provisiones.

Además, es necesario que trabaje de forma simultánea con el módulo de asistencia, a fin de que se descuenten los días de permiso, para los casos que apliquen.

La información de días laborables, del rol de pagos, se utilizará de referencia para el pago de utilidades.

Reportes: Rol de Pagos General, Rol de Pagos Individual, Pago de Utilidades

## Plantilla de Rubros

Nombre Rubro	Signo	Estado	Fecha Pago (dd/mm)
Ingreso Trabajo Continuo	Informativo	Activo	
Ingreso Variable Asignado	Informativo	Activo	
Ingreso Productividad MOD	Informativo	Activo	
Sueldo Base	Informativo	Activo	
Sueldo Básico Gobierno	Informativo	Activo	
Utilidades Recibidas	Informativo	Activo	01/04
Consumo comedor empleado	Informativo	Activo	
Valor Recargo Nocturno 25%	Ingreso	Activo	
Valor Horas Suplementarias 50%	Ingreso	Activo	
Valor Horas Extraordinarias 100%	Ingreso	Activo	
Anticipo Quincena	Egreso	Activo	
Retención Judicial	Egreso	Activo	
Retención Judicial Décimo Cuarto Mensual	Egreso	Activo	
Retención Judicial Décimo Tercero Mensual	Egreso	Activo	
Ret. Judicial XIV Sierra Acum.	Egreso	Activo	01/07
Ret. Judicial XIII Acumulado	Egreso	Activo	01/11
Préstamo Quirografario IESS	Egreso	Activo	
Préstamo Hipotecario IESS	Egreso	Activo	
Base Aportable IESS	Informativo	Activo	
Aporte Personal IESS	Egreso	Activo	
Afiliación Conyugue	Informativo	Activo	
Aporte Patronal IESS	Informativo	Activo	
Provisión IECE y CNCF	Informativo	Activo	
Provisión Décimo Tercer Sueldo	Informativo	Activo	
Décimo Tercero Mensual	Ingreso	Activo	
Décimo Tercero Acumulado	Ingreso	Activo	01/11
Provisión Décimo Cuarto Sueldo	Informativo	Activo	
Décimo Cuarto Mensual	Ingreso	Activo	
Décimo Cuarto Acumulado Sierra	Ingreso	Activo	01/07
Provisión Jubilación Patronal	Informativo	Activo	
Provisión Beneficios Patronales	Informativo	Activo	
Provisión Desahucio	Informativo	Activo	
Provisión Vacaciones	Informativo	Activo	

Provisión Vacaciones Adicionales	Informativo	Activo	
Vacaciones Mensual	Ingreso	Activo	
Provisión Fondos de Reserva	Informativo	Activo	
Fondos de Reserva Mensuales	Ingreso	Activo	
Ret. Jud. Décimo Tercer Sueldo	Informativo	Activo	01/12
Ret. Jud. Décimo Cuarto Sierra	Informativo	Activo	01/08
Décimo Tercer Sueldo	Informativo	Activo	01/12
Décimo Cuarto Sierra	Informativo	Activo	01/08

**Tabla:** 6. Plantilla de Rubros  
**Elaborado por:** Jorge González

### 3. Módulo Asistencia

Para generar la información de la asistencia, es necesario extraer de los relojes biométricos los datos y cargarlos al sistema, de tal forma que se pueda determinar el tiempo extra, descuentos por atrasos e inasistencias.

La Empresa trabaja de forma continua, en turnos rotativos de 8 horas.

<b>CONCLUSIONES ADICIONALES:</b>					
<b>RESOLUCIONES PENDIENTES PARA LA SIGUIENTE REUNION</b>					
		<b>P = PENDIENTE</b>		<b>C = CONCLUIDO</b>	
<b>#</b>	<b>RESOLUCION</b>	<b>RESPONSABLE</b>	<b>P</b>	<b>C</b>	<b>FECHA</b>
1	Elaborar los walktrough	Jorge González	X		09.06.2017

### 2. ACTA DE REUNIÓN 003



## Acta de Reuniones

<b>Motivo: Levantar información para elaboración de walkthrough</b>	<b>Acta N. 003</b>	<b><u>Versión: 1</u></b>
<b>Lugar de reunión: Prodegel</b>		
<b>Fecha de reunión: 09.06.2017</b>		
<b>Hora de reunión: 08:00 – 16:00</b>		

**Funcionarios fijos a la reunión: P = PRESENTE    A = AUSENTE**

<b>Nombre</b>	<b>Cargo</b>	<b>Rol</b>	
Fabricio Palacios	Gerente General	Sponsor del Proyecto	P
Pamela Morales	Analista de Recursos Humanos	Experto de Dominio	P
Jorge González		Administrador del Proyecto	P

## **ORDEN DEL DIA:**

<b>1. Revisión de las walkthrough</b>
<b>2.- Procesos Empleados</b>
<b>3.- Procesos Nómina</b>
<b>4.- Procesos Asistencia</b>

## **PUNTOS TRATADOS:**

### **1. Revisión de walkthroughs**

Se presentan los documentos, los mismos que son aprobados sin requerir de ajustes, el tiempo de trabajo dentro de la Empresa en el Área de Tecnología, hace más comprensibles los requerimientos del usuario y procesos a ejecutarse.

### **2. Procesos Empleados**

#### a) Creación de personas

Una vez contratado el colaborador, debe ser creado en el sistema, esto tanto para los empleados de Prodegel como de las empresas complementarias, por eso la necesidad de poder diferenciar la Empresa para la que trabajan. Los datos a insertar son los indicados en el walkthrough.

Debe permitirse la actualización de los datos de las personas, sin embargo, se debe controlar la eliminación.

#### b) Registro de contratos

Una vez creada la persona, se crea el contrato fuera del sistema en las plantillas de Word existentes, luego se registrará el contrato en el *software*, con todos los datos especificados en el walkthrough. Mientras la persona tenga un contrato activo, la misma forma parte del rol de pagos del mes. Los tipos de contrato deben ser configurables, debido a que pueden existir cambios en la normativa legal.

c) Añadir cargas familiares

Con los documentos físicos de respaldo correspondientes, se registran las cargas familiares para pago de utilidades, de requerir se pueden incluir familiares que no son cargas familiares, para ello es importante incluir un campo en el que se pueda marcar si es o no carga.

d) Actualización de contratos

Debido a que los colaboradores, pueden sufrir varios cambios de cargo, área, etc. durante su tiempo de trabajo de Prodegel, bajo el mismo contrato, es necesario que este permita actualizaciones.

e) Finalización de contrato

Al concluir un contrato es importante el poder incluir la fecha de salida, el motivo (para entes de Gobierno) y la causa (para control interno de la Empresa).

### 3. Procesos de Nómina

#### a) Creación de rol de pagos

Todos los meses se crea un rol de pagos, el mismo debe incluir a todo el personal activo y de cada empleado los rubros asignados. Al crear un rol de pagos, se debe especificar la fecha de inicio y de fin del mismo, por control interno, no se debe crear un nuevo rol, si el anterior no fue cerrado, para ello el rol debe tener un status.

#### b) Creación y Actualización de rubros

Todos los rubros que forman parte del rol de pagos, deben ser creados y configurados, pueden existir rubros fijos y rubros variables, dependiendo de la fórmula de cálculo.

Para configurar un rubro, se debe tener en cuenta lo siguiente:

- Código de rubro
- Nombre del rubro
- Status Activo (si/no)
- Tipo de rubro (informativo, ingreso, egreso, provisión)
- Dependiente de días laborables (si/no)
- Pago (mensual/anual)
- Rubros de los que depende su cálculo (Origen)
- Mes de pago
- Días de gracia
- Factor de cálculo

Para el registro de los rubros por empleado, es necesario que exista una pantalla en donde registrarlos.

Los cálculos de los rubros, se realizarán internamente en el código, no estarán a disponibilidad del usuario los cambios en las fórmulas.

Al crear la nómina, es importante tener en cuenta que las fechas para cálculo de asistencia son diferentes a las de la nómina.

#### **4. Procesos de Asistencia**

##### **a) Registros de asistencia**

La asistencia se genera de los registros de los relojes biométricos, por lo que se debe buscar una alternativa para cargarlos al sistema, a fin de que se puedan hacer los cálculos en función de la misma.

Cada registro de asistencia, cuenta con:

- Código del empleado para registro en los relojes
- Tipo de registro (entrada / Salida)
- Hora

##### **b) Tiempo extra**

El tiempo extra, se calcula en función de la hora de entrada y salida de la persona, dependiendo del día y la hora, existen los siguientes tipos de hora extra:

- 25 % Recargo nocturno (horas de trabajo a partir de las
- 50% Horas suplementarias (tiempo extra de lunes a viernes)
- 100% Horas extraordinarias (fin de semana y feriados)

Los pagos de horas extras, son aprobados por el jefe inmediato en función de lo que establece el código de trabajo.

No todo el personal tiene derecho a horas extras, por ser personal de confianza, de acuerdo al código del trabajo.

**c) Jornada de Trabajo**

El personal operativo trabaja en turnos rotativos, continuos, de la siguiente forma:

6H45 A 15H15

14H45 A 23H15

22H45 A 07h15

El personal administrativo trabaja de lunes a viernes, de 08:00 a 16:30.

**d) Permisos**

Los permisos se aplican según el código del trabajo, pueden ser:

- Permiso médico
- Permiso para estudios
- Permiso por calamidad doméstica
- Permiso por maternidad
- Permiso por paternidad
- Permiso remunerado

Los permisos son aprobados por el jefe inmediato, y son registrados por el personal del Área de Talento Humano, aquí se presenta el cambio al poder ser registrados por el empleado.

**e) Anticipo de vacaciones**

Los anticipos de vacaciones los podrá registrar el personal del área de Talento Humano o el empleado, y es aprobado por la Jefatura de Recursos Humanos. Al existir la aprobación, se debe generar un documento de respaldo que se imprime para que el colaborador y el jefe inmediato lo firmen.

**f) Vacaciones**

Todo empleado tiene derecho a vacaciones, de acuerdo al Código del Trabajo, de los días que tenga derecho se descuentan los días de anticipo de vacaciones.

La solicitud de vacaciones debe ser registrada por el empleado y aprobada por el jefe de Inmediato, de dicha aprobación se genera un documento de respaldo que debe ser firmado por el jefe inmediato y el empleado.

A partir del quinto año de laborar en la Empresa, el colaborador tiene derecho a un día extra de vacaciones por cada año, hasta un límite de 30 días.

<b>CONCLUSIONES ADICIONALES:</b>					
<b>RESOLUCIONES PENDIENTES PARA LA</b>					
<b>SIGUIENTE REUNION      P = PENDIENTE    C= CONCLUIDO</b>					
#	RESOLUCION	RESPONSABLE	P	C	FECHA

**Anexo 3: Desarrollo por características**

A continuación, el detalle del código fuente de las clases creadas:

### Clase países

```
<?php
class Paises_model extends CI_Model {

    function __construct() {
        parent::__construct();
    }
    public function sel_paises(){
        $query = $this->db->query(" SELECT p.id, p.nombre_pais, p.activo

                                FROM paises p

                                ORDER BY p.nombre_pais");
        $result = $query->result();
        return $result;
    }
    public function sel_pais_id($pais){
        $query = $this->db->query("SELECT id, nombre_pais, activo
                                FROM paises WHERE id = $pais");
        $result = $query->result();
        return $result[0];
    }
    public function upd_paises($pais, $nombre, $activo){
        $query = $this->db->query(" UPDATE paises SET
                                nombre_pais = '$nombre',
                                activo = $activo
                                WHERE id = $pais");
    }
    public function add_paises($nombre, $activo){
        $query = $this->db->query("INSERT INTO paises (nombre_pais,
        activo)
                                VALUES('$nombre', $activo);");
    }
    public function candel_pais($pais){
        $query = $this->db->query("SELECT count(*) as cant FROM provincia
        WHERE id_pais = $pais");
        $result = $query->result();
        if ($result[0]->cant == 0)
            { return 1; }
        else
            { return 0; }
    }
    public function del_pais($pais){
        if ($this->candel_pais($pais) == 1){
```

```

        $query = $this->db->query("DELETE FROM paises WHERE id =
        $pais");
        return 1;
    } else {
        return 0;
    }
}
}
}

```

## Clase Provincias

<?php

```

class Provincia_model extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    public function sel_provincia(){
        $query = $this->db->query(" SELECT c.id, c.nombre_provincia,
        c.id_pais, c.activo, p.nombre_pais
        FROM provincia c
        LEFT JOIN paises p on p.id = c.id_pais
        ORDER BY c.nombre_provincia");
        $result = $query->result();
        return $result;
    }

    public function sel_provincia_id($provincia){
        $query = $this->db->query("SELECT id, nombre_provincia, id_pais,
        activo
        FROM provincia WHERE id = $provincia");
        $result = $query->result();
        return $result[0];
    }

    public function upd_provincia($provincia, $nombre, $pais, $activo){
        if (!$pais || (trim($pais) == "")) { $pais = 'NULL'; }

        $query = $this->db->query(" UPDATE provincia SET
        nombre_provincia = '$nombre',
        id_pais = $pais,
        activo = $activo
        WHERE id = $provincia");
    }

    public function add_provincia($nombre, $pais, $activo){
        if (!$pais || (trim($pais) == "")) { $pais = 'NULL'; }
    }
}

```

```

    $query = $this->db->query("INSERT INTO provincia
(nombre_provincia, id_pais, activo)
VALUES('$nombre', $pais, $activo);");
}

public function candel_provincia($provincia){
    $query = $this->db->query("SELECT count(*) as cant FROM ciudad
WHERE id_provincia = $provincia");
    $result = $query->result();

    if ($result[0]->cant == 0)
    { return 1; }
    else
    { return 0; }
}

public function del_provincia($provincia){
    if ($this->candel_provincia($provincia) == 1){
        $query = $this->db->query("DELETE FROM provincia WHERE id =
$provincia");
        return 1;
    } else {
        return 0;
    }
}

public function lst_pais(){
    $query = $this->db->query("SELECT id, nombre_pais
FROM paises WHERE activo = 1
ORDER BY nombre_pais");
    $r = $query->result();
    return $r;
}
}

```

### **Clase ciudades**

```

<?php

class Ciudad_model extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    public function sel_ciudad(){

```

```

$query = $this->db->query(" SELECT c.id, c.nombre_ciudad,
c.id_provincia, c.activo, p.nombre_provincia
FROM ciudad c
LEFT JOIN provincia p on p.id = c.id_provincia
ORDER BY c.nombre_ciudad");
$result = $query->result();
return $result;
}

public function sel_ciudad_id($ciudad){
$query = $this->db->query("SELECT id, nombre_ciudad, id_provincia,
activo
FROM ciudad WHERE id = $ciudad");
$result = $query->result();
return $result[0];
}

public function upd_ciudad($ciudad, $nombre, $provincia, $activo){
if (!$provincia) || (trim($provincia) == "") { $provincia = 'NULL'; }

$query = $this->db->query(" UPDATE ciudad SET
nombre_ciudad = '$nombre',
id_provincia = $provincia,
activo = $activo
WHERE id = $ciudad");
}

public function add_ciudad($nombre, $provincia, $activo){
if (!$provincia) || (trim($provincia) == "") { $provincia = 'NULL'; }
$query = $this->db->query("INSERT INTO ciudad (nombre_ciudad,
id_provincia, activo)
VALUES('$nombre', $provincia, $activo);");
}

public function candel_ciudad($ciudad){
$query = $this->db->query("SELECT count(*) as cant FROM ciudad
WHERE id = $ciudad");
$result = $query->result();

return 1;
}

public function del_ciudad($ciudad){
if ($this->candel_ciudad($ciudad) == 1){
$query = $this->db->query("DELETE FROM ciudad WHERE id =
$ciudad");
return 1;
} else {
return 0;
}
}

```

```

    }
}

public function lst_provincia(){
    $query = $this->db->query("SELECT id, nombre_provincia
        FROM provincia WHERE activo = 1
        ORDER BY nombre_provincia");
    $r = $query->result();
    return $r;
}
}

```

### Clase tipo de trabajador

```

<?php
class Tipotrabajador_model extends CI_Model {

    function __construct() {
        parent::__construct();
    }
    public function sel_tipotrabajador(){
        $query = $this->db->query(" SELECT t.id, t.tipo_trabajador,
t.descripcion, t.activo
        FROM tipotrabajador t
        ORDER BY t.tipo_trabajador");
        $result = $query->result();
        return $result;
    }
    public function sel_tipotrabajador_id($tipotrabajador){
        $query = $this->db->query("SELECT id, tipo_trabajador, descripcion,
activo
        FROM tipotrabajador WHERE id = $tipotrabajador");
        $result = $query->result();
        return $result[0];
    }
    public function upd_tipotrabajador($idtipotrabajador, $tipotrabajador,
$descripcion, $activo){
        if ((!$idtipotrabajador) || (trim($idtipotrabajador) == "")) {
            $idtipotrabajador = 'NULL'; }

        $query = $this->db->query(" UPDATE tipotrabajador SET
            tipo_trabajador = '$tipotrabajador',
            descripcion = '$descripcion',
            activo = $activo
            WHERE id = $idtipotrabajador");
    }
}

```

```

    public function add_tipotrabajador($tipotrabajador, $descripcion,
$activo){
    if (!$tipotrabajador || (trim($tipotrabajador) == "")) { $tipotrabajador =
'NULL'; }
    $query = $this->db->query("INSERT INTO tipotrabajador
(tipo_trabajador, descripcion, activo)
VALUES('$tipotrabajador', '$descripcion',
$activo);");
    }
    public function cant_tipotrabajador($tipotrabajador){
    $query = $this->db->query("SELECT count(*) as cant FROM
tipotrabajador WHERE id = $ciudad");
    $result = $query->result();

    return 1;
    }
    public function del_tipotrabajador($tipotrabajador){
    $query = $this->db->query("DELETE FROM tipotrabajador WHERE id
= $tipotrabajador");
    return 1;
    }
    public function lst_tipotrabajador(){
    $query = $this->db->query("SELECT id, tipo_trabajador, descripcion
FROM tipotrabajador WHERE activo = 1
ORDER BY tipo_trabajador");
    $r = $query->result();
    return $r;
    }
}
}

```

### **Clase departamento**

```

p
class Departamento_model extends CI_Model {

    function __construct() {
        parent::__construct();
    }
    public function sel_departamento(){
        $query = $this->db->query(" SELECT d.id, d.nombre_departamento,
d.id_jefedepartamento, d.activo,
textcat(textcat(e.apellidos,text ' '),e.nombres) as
nombre_empleado
FROM departamento d
LEFT JOIN empleado e on e.id_empleado =
d.id_jefedepartamento
ORDER BY d.nombre_departamento");
        $result = $query->result();
    }
}

```

```

return $result;
}
public function sel_departamento_id($departamento){
    $query = $this->db->query("SELECT id, nombre_departamento,
id_jefedepartamento, activo
        FROM departamento WHERE id = $departamento");
    $result = $query->result();
    return $result[0];
}
public function upd_departamento($departamento, $nombre, $idjefe,
$activo){
    if (($idjefe == NULL) || ($idjefe == "")) { $idjefe = 'NULL'; }
    $query = $this->db->query(" UPDATE departamento SET
        nombre_departamento = '$nombre',
        id_jefedepartamento = $idjefe,
        activo = $activo
        WHERE id = $departamento");
}
public function add_departamento($nombre, $idjefe, $activo){
    $query = $this->db->query("INSERT INTO departamento
(nombre_departamento, id_jefedepartamento, activo)
        VALUES('$nombre', $idjefe, $activo);");
}
public function candel_departamento($departamento){
    $query = $this->db->query("SELECT count(*) as cant FROM empleado
WHERE id = $departamento");
    $result = $query->result();
    if ($result[0]->cant == 0)
        { return 1; }
    else
        { return 0; }
}
public function del_departamento($departamento){
    if ($this->candel_departamento($departamento) == 1){
        $query = $this->db->query("DELETE FROM departamento WHERE
id = $departamento");
        return 1;
    } else {
        return 0;
    }
}
public function lst_empleado($departamento = 0){
    $query = $this->db->query("SELECT id_empleado,
textcat(textcat(apellidos,text ' '),nombres) as nombre_empleado
        FROM empleado WHERE
id_departamento=$departamento AND perfil = 2 AND activo = 1
        ORDER BY nombre_empleado");
    $r = $query->result();
    return $r;
}

```

```
}  
}
```

## Clase cargos

```
<?php  
class Cargo_model extends CI_Model {  
  
    function __construct() {  
        parent::__construct();  
    }  
    public function sel_cargo(){  
        $query = $this->db->query(" SELECT p.id, p.nombre_cargo, p.activo  
                                FROM cargo p  
                                ORDER BY p.nombre_cargo");  
        $result = $query->result();  
        return $result;  
    }  
  
    public function sel_cargo_id($carga){  
        $query = $this->db->query("SELECT id, nombre_cargo, activo  
                                FROM cargo WHERE id = $carga");  
        $result = $query->result();  
        return $result[0];  
    }  
  
    public function upd_cargo($carga, $nombre, $activo){  
        $query = $this->db->query(" UPDATE cargo SET  
                                nombre_cargo = '$nombre',  
                                activo = $activo  
                                WHERE id = $carga");  
    }  
  
    public function add_cargo($nombre, $activo){  
        $query = $this->db->query("INSERT INTO cargo (nombre_cargo,  
activo)  
                                VALUES('$nombre', $activo);");  
    }  
  
    public function candel_cargo($carga){  
        $query = $this->db->query("SELECT count(*) as cant FROM empleado  
WHERE id_cargo = $carga");  
        $result = $query->result();  
  
        if ($result[0]->cant == 0)  
            { return 1; }  
        else
```

```

        { return 0; }
    }

    public function del_cargo($cargo){
        if ($this->candel_cargo($cargo) == 1){
            $query = $this->db->query("DELETE FROM cargo WHERE id =
            $cargo");
            return 1;
        } else {
            return 0;
        }
    }
}

```

### Clase trabajador

```

<?php
class Empleado_model extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    public function lst_tipoidentificacion() {
        $query = $this->db->query("SELECT id_identificacion,
        desc_identificacion, cod_identificacion
        FROM identificacion;");
        $result = $query->result();
        return $result;
    }

    /* lista de empleados */
    public function lst_empleado() {
        $query = $this->db->query("SELECT e.id_empleado, e.nombres,
        e.apellidos, e.nro_ident, e.tipo_identificacion,
        e.perfil, e.telf_empleado, e.correo_empleado,
        e.activo, e.id_departamento,
        i.desc_identificacion, d.nombre_departamento
        FROM empleado e
        INNER JOIN identificacion i on i.id_identificacion =
        e.tipo_identificacion
        LEFT JOIN departamento d on d.id =
        e.id_departamento
        ORDER BY e.apellidos, e.nombres;");
        $result = $query->result();
        return $result;
    }

    public function asi_empleado($empleado) {

```

```

$query = $this->db->query("SELECT a.id, a.fecha, a.entrada_trabajo,
a.salida_trabajo,
                                a.salida_almuerzo,      a.entrada_almuerzo,
a.codigoreloj,
                                a.id_empleado,      e.nombres,      e.apellidos,
e.nro_ident
                                FROM asistencia a
                                INNER JOIN empleado e on e.id_empleado =
a.id_empleado
                                Where a.id_empleado = '$empleado'
                                ORDER BY a.fecha;");
$result = $query->result();
return $result;
}

```

```

public function upd_empleado($idempleado, $nombre, $apellido,
$tipoident, $identificacion, $perfil, $telefono, $celular, $correo,
                                $activo,      $departamento,      $lugarexpedicion,
$cedulamilitar,$profesion, $pasaporte, $fechanac,
                                $sexo,      $estadocivil,      $peso,      $stalla,      $codigoreloj,
$scalleprincipal, $numerovivienda,
                                $scalletransversal,      $sector,      $referenciavivienda,
$ciudad, $tipovivienda, $vivefamiliares,
                                $banco,      $tipocuenta,      $numerocuenta,
$nombrecontacto, $direccioncontacto,
                                $parentescocontacto, $telefonocontacto, $empresa,
$tiposangre, $tipodiscapacidad,
                                $p100discapacidad, $contrato, $cargo, $tipocontrato,
$fechaingreso, $fechasalida, $sueldo,
                                $idjornada, $causasalida){

    if (!$perfil) || (trim($perfil) == "") { $perfil = 'NULL'; }
    if (!$departamento) || (trim($departamento) == "") { $departamento =
'NULL'; }
    if (!$fechanac) || (trim($fechanac) == "") { $fechanac = 'NULL'; }
    else { $fechanac = "to_date(" . $fechanac . ", 'YYYY-MM-DD)"; }
    if (!$estadocivil) || (trim($estadocivil) == "") { $estadocivil = 'NULL'; }
    if (!$ciudad) || (trim($ciudad) == "") { $ciudad = 'NULL'; }
    if (!$tipovivienda) || (trim($tipovivienda) == "") { $tipovivienda = 'NULL';
}

    if (!$banco) || (trim($banco) == "") { $banco = 'NULL'; }
    if (!$tipocuenta) || (trim($tipocuenta) == "") { $tipocuenta = 'NULL'; }
    if (!$parentescocontacto) || (trim($parentescocontacto) == "") {
$parentescocontacto = 'NULL'; }
    if (!$empresa) || (trim($empresa) == "") { $empresa = 'NULL'; }
    if (!$tiposangre) || (trim($tiposangre) == "") { $tiposangre = 'NULL'; }
    if (!$tipodiscapacidad) || (trim($tipodiscapacidad) == "") {
$tipodiscapacidad = 'NULL'; }
}

```

```

    if ((!$p100discapacidad) || (trim($p100discapacidad) == "")) {
    $p100discapacidad = 0; }
    if ($contrato == NULL) || (trim($contrato) == "") { $contrato = 'NULL'; }
    if (!$cargos) || (trim($cargos) == "") { $cargos = 'NULL'; }
    if (!$tipocontrato) || (trim($tipocontrato) == "") { $tipocontrato = 'NULL';
}
    if (!$fechaingreso) || (trim($fechaingreso) == "") { $fechaingreso =
'NULL'; }
    else { $fechaingreso = "to_date(" . $fechaingreso . ", 'YYYY-MM-
DD)"; }
    if (!$fechasalida) || (trim($fechasalida) == "") { $fechasalida = 'NULL'; }
    else { $fechasalida = "to_date(" . $fechasalida . ", 'YYYY-MM-DD)";
}
    if (!$sueldo) || (trim($sueldo) == "") { $sueldo = 0; }
    if (!$peso) || (trim($peso) == "") { $peso = 'NULL'; }
    if (!$talla) || (trim($talla) == "") { $talla = 'NULL'; }
    if (!$causasalida) || (trim($causasalida) == "") { $causasalida = "; }
    if (!$idjornada) || (trim($idjornada) == "") { $idjornada = 'NULL'; }

```

```

$this->db->query(" UPDATE empleado SET
    apellidos = '$apellido',
    nombres = '$nombre',
    tipo_identificacion = $tipoident,
    nro_ident = '$identificacion',
    perfil = $perfil,
    telf_empleado = '$telefono',
    celular_empleado = '$celular',
    correo_empleado = '$correo',
    id_departamento = $departamento,
    activo = $activo,
    lugarexpedicion = '$lugarexpedicion',
    cedulamilitar = '$cedulamilitar',
    profesion = '$profesion',
    pasaporte = '$pasaporte',
    fecha_nacimiento = $fechanac,
    sexo = '$sexo',
    id_estadocivil = $estadocivil,
    peso = $peso,
    talla = $talla,
    codigoreloj = '$codigoreloj',
    calleprincipal = '$calleprincipal',
    numerovivienda = '$numerovivienda',
    calletransversal = '$calletransversal',
    sector = '$sector',
    referenciavivienda = '$referenciavivienda',
    id_ciudad = $ciudad,
    id_tipovivienda = $tipovivienda,
    vivefamiliares = $vivefamiliares,
    id_banco = $banco,

```

```

id_tipocuenta = $tipocuenta,
numerocuenta = '$numerocuenta',
nombrecontacto = '$nombrecontacto',
direccioncontacto = '$direccioncontacto',
id_parentescocontacto = $parentescocontacto,
telefonocontacto = '$telefonocontacto',
id_empresa = $empresa,
id_tiposangre = $tiposangre,
id_tipodiscapacidad = $tipodiscapacidad,
p100discapacidad = $p100discapacidad,
id_contrato = $contrato,
id_cargo = $cargo,
fecha_ingreso = $fechaingreso,
fecha_salida = $fechasalida,
causa_salida = '$causasalida',
id_jornada = $idjornada,
sueldo = $sueldo
WHERE id_empleado = $idempleado");

```

```

$usua = $this->session->userdata('usua');
$idusu = $usua->id_usu;
$this->db->query("DELETE FROM cargafamiliar WHERE
id_empleado = $idempleado");
$this->db->query("INSERT INTO cargafamiliar (id_empleado,
apellidos_familiar, nombres_familiar, nro_ident,
tipo_parentesco, telf_familiar, activo,
sexo, fecha_nacimiento, fecha_fallece)
SELECT $idempleado, apellidos_familiar,
nombres_familiar, nro_ident,
tipo_parentesco, telf_familiar, activo, sexo,
fecha_nacimiento, fecha_fallece
FROM cargafamiliar_tmp t
INNER JOIN empleado_tmp e on e.id =
t.id_empleadotmp
WHERE e.id_usuario = $idusu");

```

```

if ($contrato != 'NULL'){
if ($contrato < 1){
$this->db->query("INSERT INTO contrato (id_tipo, id_empleado,
id_cargo, fecha_inicio, fecha_fin, sueldo, activo, causa_salida)
VALUES($tipocontrato, $idempleado, $cargo,
$fechaingreso, $fechasalida, $sueldo, 1, '$causasalida')");
$query = $this->db->query("SELECT max(id) as newid from
contrato;");
$result = $query->result();
$newid = $result[0]->newid;
$this->db->query("UPDATE empleado set id_contrato = $newid
WHERE id_empleado = $idempleado");
}
}

```

```

else {
    $this->db->query("UPDATE contrato SET
        id_tipo = $tipocontrato,
        id_cargo = $cargo,
        fecha_inicio = $fechaingreso,
        fecha_fin = $fechasalida,
        causa_salida = '$causasalida',
        sueldo = $sueldo
        WHERE id = $contrato");
    }
}

$this->db->query("DELETE FROM rubro_empleado WHERE
id_empleado = $idempleado");
$this->db->query("INSERT INTO rubro_empleado (id_empleado,
id_rubro, valor_net)
SELECT $idempleado, t.id_rubro, t.valor_net)
FROM rubro_empleado_tmp t
INNER JOIN empleado_tmp e on e.id =
t.id_empleadotmp
WHERE e.id_usuario = $idusu AND existe = 1");
}

public function add_empleado($nombre, $apellido, $tipoident,
$identicacion, $perfil, $telefono, $celular, $correo,
    $activo, $departamento, $lugarexpedicion,
$cedulamilitar,$profesion, $pasaporte, $fechanac,
    $sexo, $estadocivil, $peso, $talla, $codigoreloj,
$scalleprincipal, $numerovivienda,
    $scalletransversal, $sector, $referenciavivienda,
$ciudad, $tipovivienda, $vivefamiliares,
    $banco, $tipocuenta, $numerocuenta,
$nombrecontacto, $direccioncontacto,
    $parentescocontacto, $telefonocontacto, $empresa,
$tiposangre, $tipodiscapacidad,
    $p100discapacidad, $contrato, $cargo, $tipocontrato,
$fechaingreso, $fechasalida, $sueldo,
    $idjornada, $causasalida){

    if (!$perfil) || (trim($perfil) == "") { $perfil = 'NULL'; }
    if (!$departamento) || (trim($departamento) == "") { $departamento =
'NULL'; }
    if (!$fechanac) || (trim($fechanac) == "") {
        $fechanac = 'NULL';
    }
    else {
        $fechanac = "to_date('' . $fechanac . ', 'YYYY-MM-DD)";
    }
}

```

```

    if (!$estadocivil) || (trim($estadocivil) == "") { $estadocivil = 'NULL'; }
    if (!$ciudad) || (trim($ciudad) == "") { $ciudad = 'NULL'; }
    if (!$tipovivienda) || (trim($tipovivienda) == "") { $tipovivienda =
'NULL'; }
    if (!$banco) || (trim($banco) == "") { $banco = 'NULL'; }
    if (!$tipocuenta) || (trim($tipocuenta) == "") { $tipocuenta = 'NULL'; }
    if (!$parentescocontacto) || (trim($parentescocontacto) == "") {
$parentescocontacto = 'NULL'; }
    if (!$empresa) || (trim($empresa) == "") { $empresa = 'NULL'; }
    if (!$tiposangre) || (trim($tiposangre) == "") { $tiposangre = 'NULL'; }
    if (!$tipodiscapacidad) || (trim($tipodiscapacidad) == "") {
$tipodiscapacidad = 'NULL'; }
    if ($contrato == NULL) || (trim($contrato) == "") { $contrato = 'NULL';
}
    if (!$cargo) || (trim($cargo) == "") { $cargo = 'NULL'; }
    if (!$peso) || (trim($peso) == "") { $peso = 'NULL'; }
    if (!$talla) || (trim($talla) == "") { $talla = 'NULL'; }
    if (!$p100discapacidad) || (trim($p100discapacidad) == "") {
$P100discapacidad = 0; }
    if (!$tipocontrato) || (trim($tipocontrato) == "") { $tipocontrato =
'NULL'; }
    if (!$fechaingreso) || (trim($fechaingreso) == "") { $fechaingreso =
'NULL'; }
    else { $fechaingreso = "to_date(" . $fechaingreso . ", 'YYYY-MM-
DD)"; }
    if (!$fechasalida) || (trim($fechasalida) == "") { $fechasalida = 'NULL';
}
    else { $fechasalida = "to_date(" . $fechasalida . ", 'YYYY-MM-DD)";
}
    if (!$sueldo) || (trim($sueldo) == "") { $sueldo = 0; }
    if (!$causasalida) || (trim($causasalida) == "") { $causasalida = "; }
    if (!$idjornada) || (trim($idjornada) == "") { $idjornada = 'NULL'; }

```

```

$this->db->query("INSERT INTO empleado (nombres, apellidos,
tipo_identificacion, nro_ident, perfil,
                                telf_empleado,          celular_empleado,
correo_empleado, activo, id_departamento,
                                lugarexpedicion,          cedula militar, profesion,
pasaporte, fecha_nacimiento, sexo,
                                id_estadocivil, peso, talla, codigoreloj,
calleprincipal, numerovivienda,
                                calletransversal, sector, referenciavivienda,
id_ciudad, id_tipovivienda,
                                vivefamiliares, id_banco, id_tipocuenta,
numerocuenta, nombrecontacto,
                                direccioncontacto, id_parentescocontacto,
telefonocontacto, id_empresa,
                                id_tiposangre,          id_tipodiscapacidad,
p100discapacidad, id_contrato, id_cargo,

```

```

                                causa_salida, id_jornada)
                                VALUES('$nombre',          '$apellido',          $tipoident,
'Sidentificacion', $perfil, '$telefono',
                                '$celular',  '$correo',  $activo,  $departamento,
'$lugarexpedicion', '$cedulamilitar', '$profesion',
                                '$pasaporte',
                                '$fechanac,
                                '$sexo', $estadocivil, $peso, $talla,
                                '$codigoreloj', '$calleprincipal', '$numerovivienda',
'$calletransversal',
                                '$sector',          '$referenciavivienda',          $ciudad,
$tipovivienda, $vivefamiliares,
                                $banco,          $tipocuenta,          '$numerocuenta',
'$nombrecontacto', '$direccioncontacto',
                                $parentescocontacto,          '$telefonocontacto',
$empresa, $tiposangre, $tipodiscapacidad,
                                $p100discapacidad,          $contrato,          $cargo,
'$causasalida', $idjornada);");
    $query = $this->db->query("SELECT max(id_empleado) as maxid
FROM empleado");
    $resultado = $query->result();
    $newid = $resultado[0]->maxid;

    $usua = $this->session->userdata('usua');
    $idusu = $usua->id_usu;
    $this->db->query("INSERT INTO cargafamiliar (id_empleado,
apellidos_familiar, nombres_familiar, nro_ident,
                                tipo_parentesco, telf_familiar, activo,
sexo, fecha_nacimiento, fecha_fallece)
                                SELECT $newid, apellidos_familiar, nombres_familiar,
nro_ident,
                                tipo_parentesco, telf_familiar, activo, sexo,
fecha_nacimiento, fecha_fallece
                                FROM cargafamiliar_tmp t
                                INNER JOIN empleado_tmp e on e.id =
t.id_empleadotmp
                                WHERE e.id_usuario = $idusu");

    if ($contrato != 'NULL'){
        $this->db->query("INSERT INTO contrato (id_tipo, id_empleado,
id_cargo, fecha_inicio, fecha_fin, sueldo, activo, causa_salida)
                                VALUES($tipocontrato,          $newid,          $cargo,
$fechaingreso, $fechasalida, $sueldo, 1, '$causasalida')");
        $query = $this->db->query("SELECT max(id) as newid from
contrato;");
        $result = $query->result();
        $newcont = $result[0]->newid;
        $this->db->query("UPDATE empleado set id_contrato = $newcont
WHERE id_empleado = $newid");
    }

```

```

    }

}

public function del_empleado($idempleado){
    $query = $this->db->query("DELETE FROM empleado WHERE
id_empleado = $idempleado");
}

/* SELECCIONAR EL Empleado POR IDENTIF */
public function existIdentificacion($idempleado, $identificacion){
    $query = $this->db->query("SELECT count(*) as cant FROM empleado
        WHERE nro_ident = '$identificacion' and
id_empleado != $idempleado");
    $resultado = $query->result();
    return $resultado[0]->cant;
}

public function sel_empleado_id($idempleado){
    $query = $this->db->query(" SELECT e.id_empleado, e.nombres,
e.apellidos, e.nro_ident, e.tipo_identificacion,
        e.perfil, e.telf_empleado, e.celular_empleado,
e.correo_empleado,
        e.activo, e.id_departamento, e.lugarexpedicion,
e.cedulamilitar, e.profesion,
        e.pasaporte, e.fecha_nacimiento, e.sexo,
e.id_estadocivil, e.peso, e.talla,
        e.codigoreloj, e.calleprincipal,
e.numerovivienda, e.calletransversal,
        e.sector, e.referenciavivienda, e.id_ciudad,
e.id_tipovivienda, e.vivefamiliares,
        e.id_banco, e.id_tipocuenta, e.numerocuenta,
e.nombrecontacto, e.direccioncontacto,
        e.id_parentescocontacto, e.telefonocontacto,
e.id_empresa, e.id_tiposangre,
        e.id_tipodiscapacidad, e.p100discapacidad,
e.id_contrato, e.id_cargo,
        e.fecha_ingreso, e.fecha_salida, e.sueldo,
e.id_jornada, e.causa_salida,
        c.id_tipo as id_tipocontrato
        FROM empleado e
        LEFT JOIN contrato c on c.id = e.id_contrato
        WHERE e.id_empleado = $idempleado");
    $result = $query->result();
    return $result[0];
}

public function lst_parentesco() {

```

```

        $query = $this->db->query("SELECT id, parentesco FROM
parentesco;");
        $result = $query->result();
        return $result;
    }

    public function lst_estadocivil() {
        $query = $this->db->query("SELECT id, estadocivil FROM
estadocivil;");
        $result = $query->result();
        return $result;
    }

    public function lst_tipovivienda() {
        $query = $this->db->query("SELECT id, tipovivienda FROM
tipovivienda;");
        $result = $query->result();
        return $result;
    }

    public function lst_tipocuentabanco() {
        $query = $this->db->query("SELECT id, tipocuentabanco FROM
tipocuentabanco;");
        $result = $query->result();
        return $result;
    }

    public function lst_banco() {
        $query = $this->db->query("SELECT id, nombre_banco FROM banco
WHERE activo = 1;");
        $result = $query->result();
        return $result;
    }

    public function lst_sexo() {
        $query = $this->db->query("SELECT 'M' as id, 'Masculino' as sexo
UNION
SELECT 'F' as id, 'Femenino' as sexo;");
        $result = $query->result();
        return $result;
    }

    public function lst_tiposangre() {
        $query = $this->db->query("SELECT id, tiposangre FROM
tiposangre;");
        $result = $query->result();
        return $result;
    }
}

```

```

public function lst_tipodiscapacidad() {
    $query = $this->db->query("SELECT id, tipodiscapacidad FROM
tipodiscapacidad;");
    $result = $query->result();
    return $result;
}

public function lst_ciudad() {
    $query = $this->db->query("SELECT id, nombre_ciudad FROM
ciudad WHERE activo = 1;");
    $result = $query->result();
    return $result;
}

public function get_empleadotmp($usuario, $sidempleado = 0) {
    $this->db->query("DELETE FROM rubro_empleado_tmp WHERE
NOT id_empleadotmp IN
        (SELECT DISTINCT id_empleado FROM
empleado_tmp);");
    $this->db->query("DELETE FROM rubro_empleado_tmp WHERE
id_empleadotmp IN
        (SELECT id_empleado FROM empleado_tmp WHERE
id_usuario = $usuario AND id_empleado = $sidempleado);");

    $this->db->query("DELETE FROM cargafamiliar_tmp WHERE NOT
id_empleadotmp IN
        (SELECT DISTINCT id_empleado FROM
empleado_tmp);");
    $this->db->query("DELETE FROM cargafamiliar_tmp WHERE
id_empleadotmp IN
        (SELECT id_empleado FROM empleado_tmp WHERE
id_usuario = $usuario AND id_empleado = $sidempleado);");
    $this->db->query("DELETE FROM empleado_tmp WHERE
id_usuario = $usuario AND id_empleado = $sidempleado;");

    $this->db->query("INSERT INTO empleado_tmp (id_empleado,
id_usuario) Values($sidempleado, $usuario);");
    $query = $this->db->query("SELECT max(id) as newid FROM
empleado_tmp;");
    $result = $query->result();
    $newtmpid = $result[0]->newid;

    $usua = $this->session->userdata('usua');
    $idusu = $usua->id_usu;
    $this->db->query("INSERT INTO cargafamiliar_tmp
(id_empleadotmp, apellidos_familiar, nombres_familiar, nro_ident,
                    tipo_parentesco, telf_familiar, activo,
sexo, fecha_nacimiento, fecha_fallece)

```

```

SELECT          $newtmpid,          apellidos_familiar,
nombres_familiar, nro_ident,
                tipo_parentesco,   telf_familiar,   activo,   sexo,
fecha_nacimiento, fecha_fallece
FROM cargafamiliar t
WHERE id_empleado = $idempleado");

```

```

$this->db->query("INSERT INTO rubro_empleado_tmp (id_usuario,
id_empleadotmp, id_rubro, existe, valor_neto)
SELECT $idusu, $newtmpid, r.id,
CASE WHEN t.id_rubro IS NULL THEN 0 ELSE 1
END,
COALESCE(t.valor_neto, 0)
FROM rubro r
LEFT JOIN rubro_empleado t on t.id_rubro = r.id AND
t.id_empleado = $idempleado");

```

```

return $newtmpid;
}

```

```

public function del_empleadotmp($idempleadotmp) {
$this->db->query("DELETE FROM cargafamiliar_tmp WHERE
id_empleadotmp = $idempleadotmp;");
$this->db->query("DELETE FROM empleado_tmp WHERE id =
$idempleadotmp;");
}

```

```

public function sel_cargafamiliar_tmpid($idempleado){
$query = $this->db->query("SELECT t.id, t.apellidos_familiar,
t.nombres_familiar, t.nro_ident, t.tipo_parentesco,
t.telf_familiar, t.fecha_nacimiento,
t.fecha_fallece, t.sexo, t.activo, p.parentesco,
case t.sexo WHEN 'M' then 'Masculino' else
'Femenino' end as sexonombre
FROM cargafamiliar_tmp t
LEFT JOIN parentesco p on p.id = t.tipo_parentesco
WHERE id_empleadotmp = $idempleado");
$result = $query->result();
return $result;
}

```

```

public function sel_robrosempleado_tmpid($idempleado){
$query = $this->db->query("SELECT r.id, r.codigo_rubro,
r.nombre_rubro, t.valor_neto, t.existe, r.editable
FROM rubro r
INNER JOIN rubro_empleado_tmp t on t.id_rubro =
r.id
WHERE t.id_empleadotmp = $idempleado");
$result = $query->result();
return $result;
}

```

```

    }

    public function sel_cargafamiliar_id($id){
        $query = $this->db->query("SELECT id, apellidos_familiar,
nombres_familiar, nro_ident, tipo_parentesco,
telf_familiar, fecha_nacimiento, fecha_fallece,
sexo, activo
FROM cargafamiliar_tmp
WHERE id = $id");
        $result = $query->result();
        return $result[0];
    }

    /* SELECCIONAR EL Empleado POR IDENTIF */
    public function existIdentificacionCarga($idempleado, $identificacion){
        $query = $this->db->query("SELECT count(*) as cant FROM
cargafamiliar WHERE nro_ident = '$identificacion'");
        $resultado = $query->result();
        $res = $resultado[0]->cant;
        if ($res == 0){
            $query = $this->db->query("SELECT count(*) as cant FROM
cargafamiliar_tmp
WHERE nro_ident = '$identificacion' AND
id_empleadotmp = $idempleado");
            $resultado = $query->result();
            $res = $resultado[0]->cant;
        }
        return $res;
    }

    public function add_cargafamiliar_tmp($empleado, $apellidos,
$nombres, $ident, $parentesco, $telf, $fechanac, $fechafall, $activo,
$sexo){
        if (!$parentesco || (trim($parentesco) == "")) { $parentesco = 'NULL'; }
        if (!$fechanac) { $fechanac = ""; }
        if (!$fechafall) { $fechafall = ""; }
        $this->db->query("INSERT INTO cargafamiliar_tmp (id_empleadotmp,
apellidos_familiar, nombres_familiar, nro_ident,
tipo_parentesco, telf_familiar, activo,
sexo, fecha_nacimiento, fecha_fallece)
SELECT $empleado, '$apellidos', '$nombres', '$ident',
$parentesco, '$telf', $activo, '$sexo',
case when ('$fechanac' != '') then to_date('$fechanac',
'YYYY-MM-DD') else NULL end,
case when ('$fechafall' != '') then to_date('$fechafall',
'YYYY-MM-DD') else NULL end");
    }

```

```

public function upd_cargafamiliar_tmp($id, $apellidos, $nombres,
$ident, $parentesco, $telf, $fechanac, $fechafall, $activo, $sexo){
    if (!$parentesco) || (trim($parentesco) == "") { $parentesco = 'NULL'; }
    if (!$fechanac) { $fechanac = ""; }
    if (!$fechafall) { $fechafall = ""; }
    $this->db->query("UPDATE cargafamiliar_tmp SET
        apellidos_familiar = '$apellidos',
        nombres_familiar = '$nombres',
        nro_ident = '$ident',
        tipo_parentesco = $parentesco,
        telf_familiar = '$telf',
        fecha_nacimiento = case when ('$fechanac' != '') then
to_date('$fechanac', 'YYYY-MM-DD') else NULL end,
        fecha_fallece = case when ('$fechafall' != '') then
to_date('$fechafall', 'YYYY-MM-DD') else NULL end,
        activo = $activo,
        sexo = '$sexo'
        WHERE id = $id");
    }

public function lst_cargo() {
    $query = $this->db->query("SELECT id, nombre_cargo FROM cargo
WHERE activo = 1;");
    $result = $query->result();
    return $result;
}

public function lst_empresa() {
    $query = $this->db->query("SELECT id, nombre_empresa,
ruc_empresa, representante_empresa FROM empresa
        WHERE activo = 1;");
    $result = $query->result();
    return $result;
}

public function lst_tipocontrato() {
    $query = $this->db->query("SELECT id, tipocontrato FROM
tipocontrato;");
    $result = $query->result();
    return $result;
}

public function actualiza_rubroempleado($idusu, $empleadotmp, $id,
$existe, $valor) {
    $this->db->query("UPDATE rubro_empleado_tmp SET
        valor_netto = $valor,
        existe = $existe
        WHERE id_rubro = $id AND id_empleadotmp =
$empleadotmp AND id_usuario = $idusu;");
}

```



```

UNION SELECT 4 as id, 'Abril' as mes
UNION SELECT 5 as id, 'Mayo' as mes
UNION SELECT 6 as id, 'Junio' as mes
UNION SELECT 7 as id, 'Julio' as mes
UNION SELECT 8 as id, 'Agosto' as mes
UNION SELECT 9 as id, 'Septiembre' as mes
UNION SELECT 10 as id, 'Octubre' as mes
UNION SELECT 11 as id, 'Noviembre' as mes
UNION SELECT 12 as id, 'Diciembre' as mes
ORDER BY id;");

```

```
$result = $query->result();
```

```
return $result;
```

```
}
```

```
public function lst_rubro() {
```

```

    $query = $this->db->query("SELECT r.id, r.codigo_rubro,
r.nombre_rubro, r.tipo_rubro, r.afectadopordias, r.periodicidadmensual,
r.mesactivo, r.diasgracia, r.editable,
r.expresioncalculado, r.activo,

```

```
    t.tiporubro
```

```
    FROM rubro r
```

```
    INNER JOIN tiporubro t on t.id = r.tipo_rubro
```

```
    Where r.activo = 1
```

```
    ORDER BY r.codigo_rubro;");
```

```
$result = $query->result();
```

```
return $result;
```

```
}
```

```

public function add_rubro($codigo_rubro, $nombre_rubro, $rubro_activo,
$tipo_rubro, $periodo, $mesactivo, $diastrabajados, $diasgracia, $calculado,
$expresion) {

```

```
$this->db->query("INSERT INTO rubro (codigo_rubro, nombre_rubro,
tipo_rubro, activo, periodicidadmensual, mesactivo, afectadopordias,
diasgracia, editable, expresioncalculo)
```

```
VALUES ('$codigo_rubro', '$nombre_rubro', $tipo_rubro,
$rubro_activo, $periodo, $mesactivo, $diastabajados, $diasgracia, 1 -
$calculado, '$expresion')");
```

```
}
```

```
public function upd_rubro($id, $codigo_rubro, $nombre_rubro,
$rubro_activo,$tipo_rubro, $periodo, $mesactivo, $diastabajados,
$diasgracia, $calculado, $expresion) {
```

```
$this->db->query("UPDATE rubro SET
```

```
codigo_rubro = '$codigo_rubro',
```

```
nombre_rubro = '$nombre_rubro',
```

```
tipo_rubro = $tipo_rubro,
```

```
activo = $rubro_activo,
```

```
afectadopordias = $diastabajados,
```

```
periodicidadmensual = $periodo,
```

```
mesactivo = $mesactivo,
```

```
diasgracia = $diasgracia,
```

```
editable = 1 - $calculado,
```

```
expresioncalculo = '$expresion'
```

```
WHERE id = $id");
```

```
}
```

```
public function cancel_rubro($id){
```

```
$query = $this->db->query("SELECT count(*) as cant FROM
rubro_empleado WHERE id_rubro = $id");
```

```
$result = $query->result();
```

```
if ($result[0]->cant == 0)
```

```
{ return 1; }
```

```

else
    { return 0; }
}

public function del_rubro($id_rubro) {
    if ($this->candel_rubro($id_rubro) == 1){
        $query = $this->db->query("DELETE FROM rubro WHERE id =
$id_rubro");
        return 1;
    } else {
        return 0;
    }
}

public function sel_rubro_id($id){
    $query = $this->db->query("SELECT id, codigo_rubro, nombre_rubro,
tipo_rubro, afectadopordias,
                                periodicidadmensual, mesactivo, diasgracia,
editable, expresioncalculo, activo
                                FROM rubro
                                Where id = $id");
    $result = $query->result();
    return $result[0];
}

public function sel_rubro_codigo($codigo){
    $query = $this->db->query("SELECT id, codigo_rubro, nombre_rubro,
tipo_rubro, afectadopordias,
                                periodicidadmensual, mesactivo, diasgracia,
editable, expresioncalculo, activo
                                FROM rubro

```

```

        Where codigo_rubro = '$codigo');
$result = $query->result();
if ($result)
    return $result[0];
else
    return NULL;
}

}

Clase asistencia

<?php

class Asistencia_model extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    /* lista de empleados */

    public function lst_asistencia($fecha) {

        $query = $this->db->query("SELECT a.id, a.fecha, a.entrada_trabajo,
a.salida_trabajo,
                                a.salida_almuerzo, a.entrada_almuerzo,
a.codigoreloj,
                                a.id_empleado, e.nombres, e.apellidos,
e.nro_ident
                                FROM asistencia a
                                INNER JOIN empleado e on e.id_empleado =
a.id_empleado");
    }
}

```

```

        Where fecha = '$fecha'
        ORDER BY e.apellidos, e.nombres;");

$result = $query->result();
return $result;
}

public function add_asistencia($fecha, $empleado, $entrada_trabajo,
$salida_almuerzo, $entrada_almuerzo, $salida_trabajo) {
    if (!$entrada_trabajo) || (trim($entrada_trabajo) == "")
    { $entrada_trabajo = 'NULL'; }
    else
    { $entrada_trabajo = "" . $entrada_trabajo . ""; }

    if (!$salida_almuerzo) || (trim($salida_almuerzo) == "")
    { $salida_almuerzo = 'NULL'; }
    else
    { $salida_almuerzo = "" . $salida_almuerzo . ""; }

    if (!$entrada_almuerzo) || (trim($entrada_almuerzo) == "")
    { $entrada_almuerzo = 'NULL'; }
    else
    { $entrada_almuerzo = "" . $entrada_almuerzo . ""; }

    if (!$salida_trabajo) || (trim($salida_trabajo) == "")
    { $salida_trabajo = 'NULL'; }
    else
    { $salida_trabajo = "" . $salida_trabajo . ""; }

    $query = $this->db->query("SELECT codigoreloj FROM empleado
WHERE id_empleado = $empleado;");

    $result = $query->result();

```

```
$codigoreloj = $result[0]->codigoreloj;
```

```
$this->db->query("INSERT INTO asistencia (fecha, id_empleado,  
codigoreloj, entrada_trabajo, salida almuerzo,
```

```
entrada almuerzo, salida_trabajo)
```

```
VALUES ('$fecha', $empleado, '$codigoreloj',  
$entrada_trabajo, $salida_almuerzo,
```

```
$entrada_almuerzo, $salida_trabajo)");
```

```
}
```

```
public function upd_asistencia($id, $fecha, $empleado, $entrada_trabajo,  
$salida_almuerzo,
```

```
$entrada_almuerzo, $salida_trabajo) {
```

```
if (!$entrada_trabajo || (trim($entrada_trabajo) == ""))
```

```
{ $entrada_trabajo = 'NULL'; }
```

```
else
```

```
{ $entrada_trabajo = "" . $entrada_trabajo . ""; }
```

```
if (!$salida_almuerzo || (trim($salida_almuerzo) == ""))
```

```
{ $salida_almuerzo = 'NULL'; }
```

```
else
```

```
{ $salida_almuerzo = "" . $salida_almuerzo . ""; }
```

```
if (!$entrada_almuerzo || (trim($entrada_almuerzo) == ""))
```

```
{ $entrada_almuerzo = 'NULL'; }
```

```
else
```

```
{ $entrada_almuerzo = "" . $entrada_almuerzo . ""; }
```

```
if (!$salida_trabajo || (trim($salida_trabajo) == ""))
```

```
{ $salida_trabajo = 'NULL'; }
```

```
else
```

```
{ $salida_trabajo = "" . $salida_trabajo . ""; }
```

```
$query = $this->db->query("SELECT codigoreloj FROM empleado  
WHERE id_empleado = $empleado;");
```

```
$result = $query->result();
```

```
$codigoreloj = $result[0]->codigoreloj;
```

```
$this->db->query("UPDATE asistencia SET
```

```
    fecha = '$fecha',
```

```
    id_empleado = $empleado,
```

```
    codigoreloj = '$codigoreloj',
```

```
    entrada_trabajo = $entrada_trabajo,
```

```
    salida_almuerzo = $salida_almuerzo,
```

```
    entrada_almuerzo = $entrada_almuerzo,
```

```
    salida_trabajo = $salida_trabajo
```

```
WHERE id = $id");
```

```
}
```

```
public function candel_asistencia($id){
```

```
    $query = $this->db->query("SELECT count(*) as cant FROM empleado  
WHERE id_asistencia = $id");
```

```
    $result = $query->result();
```

```
    if ($result[0]->cant == 0)
```

```
    { return 1; }
```

```
    else
```

```
    { return 0; }
```

```
}
```

```
public function del_asistencia($id) {
```

```

    $this->db->query("DELETE FROM asistencia WHERE id = $id");
    return 1;
}

```

```

public function sel_asistencia_id($id){
    $query = $this->db->query("SELECT id, fecha, id_empleado, codigoreloj,
    entrada_trabajo, salida_almuerzo,
        entrada_almuerzo, salida_trabajo
        FROM asistencia
        Where id = $id");
    $result = $query->result();
    return $result[0];
}

```

```

public function lst_empleado($fecha, $sidempleado = 0) {
    $query = $this->db->query("SELECT e.id_empleado, e.nombres,
    e.apellidos, e.nro_ident
        FROM empleado e
        WHERE e.activo = 1 AND
            (e.id_empleado = $sidempleado OR
                NOT e.id_empleado IN (SELECT id_empleado
    FROM asistencia WHERE fecha = '$fecha'))
        ORDER BY e.apellidos, e.nombres;");
    $result = $query->result();
    return $result;
}
}

```

**Clase de jornada**

```
<?php
```

```
class Jornada_model extends CI_Model {
```

```
    function __construct() {  
        parent::__construct();  
    }
```

```
    /* lista de empleados */
```

```
    public function lst_jornada() {
```

```
        $query = $this->db->query("SELECT id, descripcion, entrada_trabajo,  
salida almuerzo, entrada almuerzo, salida_trabajo, activo
```

```
        FROM jornada
```

```
        Where activo = 1
```

```
        ORDER BY descripcion;");
```

```
        $result = $query->result();
```

```
        return $result;
```

```
    }
```

```
    public function add_jornada($descripcion, $entrada_trabajo,  
$salida almuerzo, $entrada almuerzo, $salida_trabajo, $activo) {
```

```
        if (!$entrada_trabajo || (trim($entrada_trabajo) == ""))
```

```
            { $entrada_trabajo = 'NULL'; }
```

```
        else
```

```
            { $entrada_trabajo = "" . $entrada_trabajo . ""; }
```

```
        if (!$salida almuerzo) || (trim($salida almuerzo) == "")
```

```
            { $salida almuerzo = 'NULL'; }
```

```
        else
```

```
            { $salida almuerzo = "" . $salida almuerzo . ""; }
```

```
if ((!$entrada_almuerzo) || (trim($entrada_almuerzo) == ""))
    { $entrada_almuerzo = 'NULL'; }
```

```
else
```

```
    { $entrada_almuerzo = "" . $entrada_almuerzo . ""; }
```

```
if ((!$salida_trabajo) || (trim($salida_trabajo) == ""))
```

```
    { $salida_trabajo = 'NULL'; }
```

```
else
```

```
    { $salida_trabajo = "" . $salida_trabajo . ""; }
```

```
$this->db->query("INSERT INTO jornada (descripcion, entrada_trabajo,
salida_almuerzo, entrada_almuerzo, salida_trabajo, activo)
```

```
VALUES ('$descripcion', $entrada_trabajo,
$salida_almuerzo, $entrada_almuerzo, $salida_trabajo, $activo)");
```

```
}
```

```
public function upd_jornada($id, $descripcion, $entrada_trabajo,
$salida_almuerzo, $entrada_almuerzo, $salida_trabajo, $activo) {
```

```
    if ((!$entrada_trabajo) || (trim($entrada_trabajo) == ""))
```

```
        { $entrada_trabajo = 'NULL'; }
```

```
    else
```

```
        { $entrada_trabajo = "" . $entrada_trabajo . ""; }
```

```
    if ((!$salida_almuerzo) || (trim($salida_almuerzo) == ""))
```

```
        { $salida_almuerzo = 'NULL'; }
```

```
    else
```

```
        { $salida_almuerzo = "" . $salida_almuerzo . ""; }
```

```
    if ((!$entrada_almuerzo) || (trim($entrada_almuerzo) == ""))
```

```
        { $entrada_almuerzo = 'NULL'; }
```

```
    else
```

```
{ $entrada_almuerzo = "" . $entrada_almuerzo . ""; }
```

```
if (!$salida_trabajo || (trim($salida_trabajo) == ""))
```

```
{ $salida_trabajo = 'NULL'; }
```

```
else
```

```
{ $salida_trabajo = "" . $salida_trabajo . ""; }
```

```
$this->db->query("UPDATE jornada SET
```

```
    descripcion = '$descripcion',
```

```
    entrada_trabajo = $entrada_trabajo,
```

```
    salida_almuerzo = $salida_almuerzo,
```

```
    entrada_almuerzo = $entrada_almuerzo,
```

```
    salida_trabajo = $salida_trabajo,
```

```
    activo = $activo
```

```
WHERE id = $id");
```

```
}
```

```
public function candel_jornada($id){
```

```
    $query = $this->db->query("SELECT count(*) as cant FROM empleado  
WHERE id_jornada = $id");
```

```
    $result = $query->result();
```

```
    if ($result[0]->cant == 0)
```

```
    { return 1; }
```

```
    else
```

```
    { return 0; }
```

```
}
```

```
public function del_jornada($id_jornada) {
```

```
    if ($this->candel_jornada($id_jornada) == 1){
```

```
        $query = $this->db->query("DELETE FROM jornada WHERE id =  
$id_jornada");
```

```
        return 1;
    } else {
        return 0;
    }
}
```

```
public function sel_jornada_id($id){
    $query = $this->db->query("SELECT id, descripcion, entrada_trabajo,
salida_almuerzo, entrada_almuerzo, salida_trabajo, activo
                                FROM jornada
                                Where id = $id");
    $result = $query->result();
    return $result[0];
}
```

```
}
```

### **Clase rol de pagos**

```
<?php
```

```
class Rol_model extends CI_Model {
```

```
    function __construct() {
        parent::__construct();
    }
```

```
    /* lista de roles */
```

```

public function lst_roles() {
    $query = $this->db->query("SELECT id, descripcion_rol, fechaini_rol,
    fechafin_rol,
                                estado_rol, asistencia_ini, asistencia_fin
    FROM roldepagos
    ORDER BY fechaini_rol DESC;");
    $result = $query->result();
    return $result;
}

public function carga_rol($idrol, $idusuario) {
    $query = $this->db->query("DELETE FROM roldepagos_tmp WHERE
    id_usuario = $idusuario;");
    $query = $this->db->query("DELETE FROM roldepagos_tmpdet
    WHERE id_usuario = $idusuario;");
    if ($idrol != 0){
        $this->db->query("INSERT INTO roldepagos_tmp (id_usuario,
    descripcion_rol, fechaini_rol, fechafin_rol,
                                estado_rol, asistencia_ini, asistencia_fin)
    SELECT $idusuario, descripcion_rol, fechaini_rol,
    fechafin_rol,
                                estado_rol, asistencia_ini, asistencia_fin
    FROM roldepagos
    WHERE id = $idrol;");
        $this->db->query("INSERT INTO roldepagos_tmpdet (id_usuario,
    id_empleado, id_rubro, valor_neto)
    SELECT $idusuario, id_empleado, id_rubro, valor_neto
    FROM roldepagos_det
    WHERE id_rol = $idrol;");
    }
}

```

```

else {
    $query = $this->db->query("SELECT count(*) as cant FROM
roldepagos");
    $result = $query->result();
    if ($result[0]->cant > 0){
        $query = $this->db->query("SELECT date(max(fechafin_rol) +
interval '1 day') as ini,
                                date(date_trunc('month', max(fechafin_rol) +
interval '1 day') + interval '1 month' - interval '1 day') as fin
                                FROM roldepagos");
        $result = $query->result();

        $inirol = $result[0]->ini;
        $finrol = $result[0]->fin;
    }
    else {
        $query = $this->db->query("SELECT date(date_trunc('month',
current_date)) as ini,
                                date(date_trunc('month', current_date) + interval
'1 month' - interval '1 day') as fin");
        $result = $query->result();
        $inirol = $result[0]->ini;
        $finrol = $result[0]->fin;
    }

    $this->db->query("INSERT INTO roldepagos_tmp (id_usuario,
descripcion_rol, estado_rol, fechaini_rol, fechafin_rol,
                                asistencia_ini, asistencia_fin)
                                VALUES ($idusuario, ' ', 1, to_date('$inirol', 'YYYY-MM-
DD'), to_date('$finrol', 'YYYY-MM-DD'),
                                to_date('$inirol', 'YYYY-MM-DD'),
to_date('$finrol', 'YYYY-MM-DD'))");
}

```

```

$this->db->query("INSERT INTO roldepagos_tmpdet (id_usuario,
id_empleado, id_rubro, valor_netto)

SELECT $idusuario, r.id_empleado, r.id_rubro, r.valor_netto

FROM empleado e

INNER JOIN rubro_empleado r on r.id_empleado =
e.id_empleado

WHERE NOT EXISTS (SELECT * FROM
roldepagos_tmpdet t

WHERE t.id_usuario = $idusuario and
t.id_empleado = r.id_empleado and
t.id_rubro = r.id_rubro);");

```

*/\* actualizando sueldo \*/*

```

$this->db->query("UPDATE roldepagos_tmpdet

SET valor_netto = empleado.sueldo

FROM empleado

WHERE roldepagos_tmpdet.id_empleado =
empleado.id_empleado AND

roldepagos_tmpdet.id_usuario = $idusuario AND

roldepagos_tmpdet.id_rubro = (SELECT valor FROM
parametros WHERE id = 2)::integer;");

```

```

$query = $this->db->query("SELECT id_usuario, descripcion_rol,
fechaini_rol, fechafin_rol,

estado_rol, asistencia_ini, asistencia_fin

FROM roldepagos_tmp

WHERE id_usuario = $idusuario;");

```

```
$result = $query->result();
```

```
$ini_asis = $result[0]->asistencia_ini;
```

```
$fin_asis = $result[0]->asistencia_fin;
```

```

/* actualizando dias trabajados */
$this->db->query("UPDATE roldepagos_tmpdet
                SET valor_neto = (SELECT count(*) FROM asistencia a
                                LEFT JOIN empleado e on e.id_empleado =
a.id_empleado
                                LEFT JOIN jornada j on j.id = e.id_jornada
                                WHERE a.id_empleado =
roldepagos_tmpdet.id_empleado AND
                                a.fecha BETWEEN '$ini_asis' AND
'$fin_asis' AND
                                ((j.id IS NULL) OR
                                (a.entrada_trabajo <= j.entrada_trabajo
AND
                                a.salida_trabajo >= j.salida_trabajo AND
                                a.salida_almuerzo >= j.salida_almuerzo
AND
                                a.entrada_almuerzo <=
j.entrada_almuerzo))
                                )
                WHERE roldepagos_tmpdet.id_usuario = $idusuario AND
                roldepagos_tmpdet.id_rubro = (SELECT valor FROM
parametros WHERE id = 3)::integer;");

return $result[0];
}

public function lst_empleados($idusuario) {
    $query = $this->db->query("SELECT e.id_empleado, e.nombres,
e.apellidos, e.nro_ident, t.valor_neto
                                FROM roldepagos_tmpdet t
                                INNER JOIN empleado e on e.id_empleado =
t.id_empleado

```

```

WHERE t.id_usuario = $idusuario AND
      t.id_rubro = (SELECT valor FROM parametros
WHERE id = 4)::integer /*neto a cobrar*/

ORDER BY e.apellidos, e.nombres;");

$result = $query->result();
return $result;
}

```

```

public function lst_rubros($idusuario, $idempleado) {
    $query = $this->db->query("SELECT t.id_rubro, r.codigo_rubro,
r.nombre_rubro, r.expresioncalculo, t.valor_neto, r.editable,
      CASE WHEN p.id IS NOT NULL then 0 else
r.editable END as modificable
      FROM roldepagos_tmpdet t
      INNER JOIN rubro r on r.id = t.id_rubro
      LEFT JOIN parametros p on p.valor = t.id_rubro::char
      WHERE t.id_usuario = $idusuario AND t.id_empleado
= $idempleado
      ORDER BY r.codigo_rubro;");

    $result = $query->result();
    return $result;
}

```

```

public function lst_rubros_calculo($idusuario, $idempleado) {
    $query = $this->db->query("SELECT r.id as id_rubro, r.codigo_rubro,
r.nombre_rubro, r.expresioncalculo,
      COALESCE(t.valor_neto,0) as valor_neto,
r.editable,
      CASE WHEN p.id IS NOT NULL then 0 else
r.editable END as modificable
      FROM rubro r

```

```

        LEFT JOIN roldepagos_tmpdet t on t.id_rubro = r.id
AND t.id_usuario = $idusuario AND t.id_empleado = $idempleado

        LEFT JOIN parametros p on p.valor = t.id_rubro::char

        ORDER BY r.codigo_rubro;");

$result = $query->result();

return $result;

}

```

```

public function actualiza_valorrubro($idusuario, $idempleado, $idrubro,
$valor) {

    if ($valor == "") { $valor = 0; }

    $this->db->query("UPDATE roldepagos_tmpdet SET valor_netto =
$valor

        WHERE id_usuario = $idusuario AND

            id_empleado = $idempleado AND

            id_rubro = $idrubro;");

}

```

```

public function sel_rubroneto_tmp($idusuario, $idempleado) {

    $query = $this->db->query("SELECT valor_netto

        FROM roldepagos_tmpdet t

        WHERE id_rubro = (SELECT valor FROM
parametros WHERE id = 4)::integer AND

            t.id_usuario = $idusuario AND t.id_empleado =
$idempleado;");

    $result = $query->result();

    if ($result)

        return $result[0]->valor_netto;

    else

        return 0;

}

```

```

public function upd_tmprol($idusuario, $fechainirol, $fechafinrol,
$feciniasist, $fecfinasist, $descripcion){

    if (!$fechainirol) || (trim($fechainirol) == "")
        { $fechainirol = 'NULL'; }
    else
        { $fechainirol = "to_date('' . $fechainirol . '' , 'YYYY-MM-DD)"; }

    if (!$fechafinrol) || (trim($fechafinrol) == "")
        { $fechafinrol = 'NULL'; }
    else
        { $fechafinrol = "to_date('' . $fechafinrol . '' , 'YYYY-MM-DD)"; }

    if (!$feciniasist) || (trim($feciniasist) == "")
        { $feciniasist = 'NULL'; }
    else
        { $feciniasist = "to_date('' . $feciniasist . '' , 'YYYY-MM-DD)"; }

    if (!$fecfinasist) || (trim($fecfinasist) == "")
        { $fecfinasist = 'NULL'; }
    else
        { $fecfinasist = "to_date('' . $fecfinasist . '' , 'YYYY-MM-DD)"; }

    $this->db->query("UPDATE roldepagos_tmp SET
        descripcion_rol = '$descripcion',
        fechaini_rol = $fechainirol,
        fechafin_rol = $fechafinrol,
        asistencia_ini = $feciniasist,

```

```

        asistencia_fin = $fecfinasist,
        estado_rol = 1
    WHERE id_usuario = $idusuario;");

}

public function add_rol($idusu){
    $this->db->query("INSERT INTO roldepagos (descripcion_rol,
    fechaini_rol, fechafin_rol,
        asistencia_ini, asistencia_fin, estado_rol)
    SELECT descripcion_rol, fechaini_rol, fechafin_rol,
        asistencia_ini, asistencia_fin, estado_rol
    FROM roldepagos_tmp
    WHERE id_usuario = $idusu");

    $query = $this->db->query("SELECT max(id) as maxid FROM
    roldepagos");
    $resultado = $query->result();
    $newid = $resultado[0]->maxid;

    $this->db->query("INSERT INTO roldepagos_det (id_rol, id_empleado,
    id_rubro, valor_neto)
        SELECT $newid, id_empleado, id_rubro, valor_neto
        FROM roldepagos_tmpdet
        WHERE id_usuario = $idusu");
}

public function upd_rol($idusu, $id){
    $this->db->query("UPDATE roldepagos SET
        descripcion_rol = t.descripcion_rol,
        fechaini_rol = t.fechaini_rol,

```

```

        fechafin_rol = t.fechafin_rol,
        asistencia_ini = t.asistencia_ini,
        asistencia_fin = t.asistencia_fin,
        estado_rol = t.estado_rol
    FROM (SELECT * FROM roldepagos_tmp WHERE
id_usuario = $idusu) t
    WHERE roldepagos.id = $id");

    $this->db->query("DELETE FROM roldepagos_det WHERE id_rol =
$id");

    $this->db->query("INSERT INTO roldepagos_det (id_rol, id_empleado,
id_rubro, valor_neto)
        SELECT $id, id_empleado, id_rubro, valor_neto
        FROM roldepagos_tmpdet
        WHERE id_usuario = $idusu");
}

public function del_rol($id){
    $this->db->query("DELETE FROM roldepagos_det WHERE id_rol =
$id");
    $this->db->query("DELETE FROM roldepagos WHERE id = $id");
}

public function lst_tmprolemp_encab($idusuario, $idempleado) {
    $query = $this->db->query("SELECT e.id_empleado, e.nombres,
e.apellidos, c.nombre_cargo,
        l. fechafin_rol,
        COALESCE(t.valor_neto,0) as diastrab
        FROM rubro r
        INNER JOIN roldepagos_tmpdet t on t.id_rubro = r.id
AND t.id_usuario = $idusuario AND t.id_empleado = $idempleado

```

```

INNER JOIN roldepagos_tmp l on l.id_usuario =
t.id_usuario

INNER JOIN parametros p on p.valor =
t.id_rubro::char

INNER JOIN empleado e on e.id_empleado =
t.id_empleado

LEFT JOIN cargo c on c.id = e.id_cargo

WHERE p.id = 3"); /*dias trabajados*/

$result = $query->result();
return $result[0];
}

public function lst_tmprolemp_rubros($idusuario, $idempleado) {
    $query = $this->db->query("SELECT r.id as id_rubro, r.codigo_rubro,
r.nombre_rubro,
                                COALESCE(t.valor_netto,0) as valor_netto,
r.tipo_rubro
                                FROM rubro r
                                LEFT JOIN roldepagos_tmplet t on t.id_rubro = r.id
AND t.id_usuario = $idusuario AND t.id_empleado = $idempleado
                                WHERE not r.id::char in (SELECT valor FROM
parametros WHERE id in (3,4)) /*excepto dias trab y neto a cobrar*/
                                ORDER BY r.codigo_rubro;");

    $result = $query->result();
    return $result;
}
}

```