



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
FACULTAD 4

# MÓDULO DE RECURSOS HUMANOS DE LA SUITE XEDRO GESPRO v15.05

*TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN  
CIENCIAS INFORMÁTICAS.*

AUTOR: Oniel González Jardínez

TUTORAS: MSc. Karina Mileisis Torres Quiñones  
Ing. Isamira Perdomo Bello

La Habana, Junio de 2015  
"Año 57 de la Revolución"

## **DECLARACIÓN DE AUTORÍA**

Por este medio declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del 2015.

Autor:

---

Oniel González Jardínez

Tutora:

---

MSc. Karina Mileisis Torres Quiñones

Tutora:

---

Ing. Isamira Perdomo Bello



**"Donde hay una empresa  
de éxito, alguien tomó  
alguna vez una  
decisión valiente."**

## AGRADECIMIENTOS



*todos mis compañeros en estos años de tristezas, alegrías y sacrificios, a los que de una forma u otra han hecho posible este momento. A la UCI Por enseñarme a vencer cualquier reto, por formarme como un Ingeniero capaz de afrontar cualquier tarea.*

*A mis amigos, por convertirse en mi familia cuando me encontraba lejos de la mía, por compartir conmigo en momentos de alegría, trabajo o sufrimiento.*

*A todos los que de una forma u otra me ayudaron en la realización de este trabajo.*

*A mi familia por apoyarme siempre y creer en mí.*

*A mis tutoras por toda su ayuda brindada, a pesar de su trabajo siempre estuvieron presente, en especial a Isamira.*

*A todos los profesores que he tenido en mi trayectoria como estudiante, cada uno de ellos me ha enseñado y han hecho posible este sueño.*

*Al mis compañeros de trabajo en GESPRO por su apoyo y ayuda que me dieron en todo momento, en especial a Henry Pestano.*

## DEDICATORIA

**A** mi mamá por ser todo para mí, por estar ahí en todo momento. Por su apoyo, dedicación y cariño que me ha dado. Por ser mi ejemplo en la vida. Por indicarme el camino bueno, pero dejarme elegir por mí mismo.

A mis hermanos por darme todo su apoyo y ayuda incondicional, en especial a Yaneisy y a Bolo.

A toda mi familia, por sus consejos y guía en toda mi carrera y en mi vida.

A mis sobrinos que los quiero mucho.

## Resumen

Las herramientas de gestión de proyectos han propiciado mejoras en los procesos de las empresas para contribuir a la toma de decisiones de forma proactiva. Con el avance de la industria del software ha sido necesario desarrollar sistemas de gestión de proyectos. Cuba, lucha por avanzar en este campo tecnológico por lo que cuenta con la Suite Xedro GESPRO desarrollada en la Universidad de las Ciencias Informáticas (UCI) la misma cuenta con un Módulo de Recursos Humanos en su versión 13.05. A partir de las vulnerabilidades detectadas en el Módulo de Recursos Humanos versión 13.05 en la presente investigación se propone una nueva versión que incorpora funcionalidades asociadas a la sobrecarga de recursos humanos para facilitar el proceso de planificación y análisis de los recursos humanos. Además se aplicaron un conjunto de pruebas para garantizar el correcto funcionamiento de la nueva versión del Módulo de Recursos Humanos y cumplir con los requisitos identificados por el cliente.

**Palabras claves:** Xedro GESPRO, Recursos Humanos, Sobrecarga

**Índice**

<b>Introducción</b> .....	<b>1</b>
<b>Capítulo 1: Fundamentación Teórica</b> .....	<b>5</b>
1.1. Conceptos asociados .....	5
1.2. Estudio comparativo de herramientas con soluciones similares .....	7
1.3. Metodología de desarrollo de software .....	12
1.4. Herramientas y tecnologías .....	15
1.5. Conclusiones parciales.....	22
<b>Capítulo 2: Propuesta de Solución</b> .....	<b>24</b>
2.1. Propuesta de solución .....	24
2.2. Especificación de requisitos .....	25
2.3. Diseño del módulo.....	29
2.4. Patrón de arquitectura .....	31
2.5. Patrones de diseño .....	32
2.6. Patrón de acceso a datos .....	34
2.7. Modelo Entidad Relación .....	35
2.8. Diagrama de componentes.....	36
2.9. Conclusiones parciales.....	37
<b>Capítulo 3. Implementación y Prueba</b> .....	<b>38</b>
3.1. Implementación del módulo .....	38
3.2. Estilos de código .....	41
3.3. Pruebas .....	44
3.4. Conclusiones parciales.....	59
<b>Conclusiones Generales</b> .....	<b>60</b>
<b>Recomendaciones</b> .....	<b>61</b>
<b>Referencias Bibliográficas</b> .....	<b>62</b>

## Índice de figuras

<b>Figura 1</b> Sistema de información transaccional Xedro GESPRO. ....	12
<b>Figura 2</b> Representación del proceso de desarrollo de la metodología Scrum.....	15
<b>Figura 3</b> Estructura del módulo de RRHH de la Suite Xedro GESPRO v13.05.....	30
<b>Figura 4</b> Nueva estructura del módulo de RRHH de la Suite Xedro GESPRO. ....	30
<b>Figura 5</b> Niveles lógicos de la arquitectura MVC.....	31
<b>Figura 6</b> Arquitectura del framework RoR, basado en el patrón MVC. ....	32
<b>Figura 7</b> Ejemplo del uso de patrón Active Record. ....	34
<b>Figura 8</b> Diagrama entidad relación del módulo de RRHH.....	35
<b>Figura 9</b> Diagrama de componentes de los módulos de RRHH. ....	37
<b>Figura 10</b> Estructura del módulo de RRHH. ....	39
<b>Figura 11</b> Diagrama de despliegue de la Suite Xedro GESPRO.....	41
<b>Figura 12</b> Ejemplo de formateo del código. ....	42
<b>Figura 13</b> Alinear la vista de presentación del módulo de RRHH a los principales procesos de la Gestión de RRHH.....	46
<b>Figura 14</b> Visualizar los recursos humanos conjuntamente con su índice de sobrecarga (IRHA).....	48
<b>Figura 15</b> Colorear el índice de sobrecarga de los RRHH.....	49
<b>Figura 16</b> Ordenar ascendentemente o descendentemente la lista de RRHH según el índice de sobrecarga. ....	51
<b>Figura 17</b> Visualizar calendario de tareas concurrentes. ....	53
<b>Figura 18</b> Calendario de tareas correspondiente a un recurso humano filtradas por mes y año. ....	56
<b>Figura 19</b> Mostrar mensaje de disponibilidad: “Disponible”, “No disponible” o “Disponible con riesgo” correspondiente a cada recurso humano de un proyecto. ....	58

**Índice de tablas**

<b>Tabla 1</b> Nuevos atributos del HTML5. Fuente: (38).....	18
<b>Tabla 2</b> Requisitos no funcionales. ....	26
<b>Tabla 3</b> Descripción de las clases controladoras del módulo de RRHH. ....	40
<b>Tabla 4</b> Descripción de la clases modelos del módulo de RRHH. ....	40
<b>Tabla 5</b> Descripción de las clases auxiliares del módulo de RRHH. ....	40
<b>Tabla 6</b> Mostrar Recursos Humanos. ....	46
<b>Tabla 7</b> Visualizar los recursos humanos conjuntamente con su índice de sobrecarga. ....	47
<b>Tabla 8</b> Colorear el índice de sobrecarga de los RRHH. ....	48
<b>Tabla 9</b> Ordenar ascendentemente o descendientemente la lista de RRHH según el índice de sobrecarga. ....	50
<b>Tabla 10</b> Visualizar calendario de tareas concurrentes por recurso humano. ....	51
<b>Tabla 11</b> Descripción de los parámetros. ....	52
<b>Tabla 12</b> Visualizar calendario de tareas concurrentes filtradas por mes y año, además por recurso humano. ....	54
<b>Tabla 13</b> Descripción de los parámetros. ....	55
<b>Tabla 14</b> Rediseño de la vista del cronograma de tareas resaltando el uso del índice de sobrecarga. ....	56
<b>Tabla 15</b> Mostrar mensaje de disponibilidad: “Disponible”, “No disponible” o “Disponible con riesgo” correspondiente a cada recurso humano de un proyecto. ....	57

## INTRODUCCIÓN

El desarrollo alcanzado y la progresiva expansión del mundo de la industria de software, ha traído consigo importantes logros científico-técnicos y un gran impacto social y económico. Esta área relativamente joven, aún se vislumbra como una industria en evolución y crecimiento. En esta industria, entre los factores fundamentales que influyen en el éxito de los proyectos, se encuentran los relacionados con los recursos humanos y las decisiones que involucra la gestión de los mismos.

Los recursos humanos los constituyen las personas y sus energías, el motor fundamental para el aprovechamiento de esas energías lo constituye la motivación. La preocupación de hoy es resolver la problemática del hombre, como recurso y como factor fundamental de la dinámica de las organizaciones y de la captación de sus energías de una manera favorable (1).

La correcta gestión de los recursos humanos teniendo en cuenta sus capacidades, disponibilidad y sobrecarga, garantizará el incremento de la productividad, calidad, seguridad, alcance de las metas establecidas y un alto grado de competitividad.

La industria cubana experimenta dificultades en el desarrollo exitoso de sus proyectos. Los recientes Lineamientos de la Política Económica y Social aprobados en el XI Congreso del Partido Comunista de Cuba (2) reflejan la urgencia de contar con mecanismos eficientes de control que permitan conocer el estado de los proyectos y las empresas, contribuyendo a la toma de decisiones de forma proactiva. En este sentido, la gestión de proyectos cobra vital importancia.

La gestión de proyectos es la disciplina que se encarga de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo y coste definidos. Entre las entidades productoras de software en Cuba se encuentra la UCI, la cual ha organizado su infraestructura productiva en centros de desarrollo de software donde se lleva a cabo todo un proceso encaminado al desarrollo de proyectos informáticos.

La UCI presenta entre sus objetivos, desarrollar proyectos de carácter informático. Entre estos se distingue, por su impacto en el ámbito social y educacional, el proyecto de Investigaciones en Gestión de Proyectos (GESPRO), perteneciente al departamento de

Gestión de Proyectos del Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE).

La Suite Xedro GESPRO permite el almacenamiento de los datos correspondientes a la gestión de los proyectos desarrollados. Esta herramienta sirve de apoyo a la gestión de proyectos informáticos en todos los centros de desarrollo de la Universidad, es utilizada por la Red de Centros Productivos de la UCI y la Empresa de Tecnologías de la Información para la Defensa (XETID) para monitorear y controlar el estado de ejecución de sus proyectos.

El Departamento de Gestión de Proyectos, la Red de Centros y la XETID, han detectado deficiencias durante la ejecución de los proyectos, específicamente en el área de Gestión de Recursos Humanos, identificando como problema fundamental la sobreasignación de tareas de los mismos, lo cual ha generado incumplimiento en los tiempos de entrega y malas evaluaciones durante los cortes de ejecución de los proyectos.

Esto se debe a que el Módulo de Recursos Humanos (RRHH) de la herramienta se encuentra muy limitado respecto a la sobrecarga de los recursos, solo brinda funcionalidades para analizar el índice de productividad individual de los miembros de un proyecto, no es posible especificar el fondo de tiempo de cada recurso humano, se realiza la planificación de recursos humanos sin tener en cuenta el fondo de tiempo y su disponibilidad, provocando sobrecarga en dicho recurso y además, no se analiza, ni evalúa el estado de explotación de un recurso, influyendo de manera negativa en el proceso de análisis de un proyecto. Los mecanismos existentes dificultan el proceso de planificación de un proyecto, ya que no se brinda información de disponibilidad de un recurso humano.

A partir de la situación problemática planteada surge el siguiente **problema a resolver**: ¿Cómo facilitar la planificación y el análisis de recursos humanos en la Suite Xedro GESPRO?

Se traza como **objeto de estudio** para la presente investigación el siguiente: proceso de planificación y análisis de recursos humanos, teniendo en cuenta la sobrecarga de recursos humanos, enmarcado en el **campo de acción**: herramientas de gestión de proyectos informáticos.

Se plantea como **objetivo general**: Desarrollar nuevas funcionalidades teniendo en cuenta la sobrecarga de recursos humanos, en el módulo de RRHH de la Suite Xedro GESPRO que facilite la planificación y el análisis de los recursos humanos.

Desglosándose el objetivo general en los siguientes **objetivos específicos**:

- ✓ Analizar el marco teórico y referencial de la investigación relacionado con la sobrecarga de los recursos humanos.
- ✓ Diseñar la arquitectura del módulo de RRHH para guiar las actividades de implementación.
- ✓ Implementar las nuevas funcionalidades del módulo de RRHH para dar cumplimiento a los requisitos definidos en el negocio.
- ✓ Validar la solución propuesta mediante aplicación de pruebas de software para el aseguramiento de la calidad del producto obtenido.

**Tareas a cumplir:**

1. Elaboración del diseño teórico y metodológico de la investigación.
2. Realización de análisis críticos y valorativos de los sistemas informáticos que realizan tareas de gestión de recursos humanos.
3. Evaluación de las herramientas o componentes que se utilizarán para el análisis y diseño del sistema.
4. Identificación y descripción de las especificaciones de software.
5. Especificación de las estrategias de codificación, los estándares y estilos a utilizar.
6. Implementación de las funcionalidades definidas en las especificaciones de software.
7. Validación de la solución propuesta mediante aplicación de pruebas de software.

La **hipótesis**, se plantea de la siguiente manera: Si se desarrollan funcionalidades teniendo en cuenta la sobrecarga de recursos humanos, entonces se facilitará la planificación y el análisis de los recursos humanos en el módulo de RRHH de la Suite Xedro GESPRO.

**Posibles resultados:**

- ✓ Documentación de acuerdo a la metodología de desarrollo de software seleccionada.
- ✓ El módulo de RRHH de la Suite Xedro GESPRO, contará con la incorporación de nuevas funcionalidades, que facilitarán la planificación y el análisis los de recursos humanos a partir de la sobrecarga de recursos humanos.

Para el desarrollo de la investigación se utilizaron los siguientes **métodos de investigación**:

**Métodos teóricos-lógicos:**

**Histórico-lógico:** para profundizar en los antecedentes y elementos que incidieron en las distintas etapas del desarrollo del módulo de RRHH en la versión 13.05.

**Analítico-sintético:** para el análisis de algunos temas teóricos que se recogen en las bibliografías consultadas. Además permitió acotar y valorar la información previamente estudiada para establecer características y conclusiones que aporten al resultado de la investigación.

**Métodos empíricos:**

**Observación:** permitió realizar un estudio de las deficiencias que trae consigo el proceso de planificación de un recurso humano sin tener en cuenta la carga de trabajo del mismo.

**Entrevista:** se le aplicó a los especialistas del Departamento de Gestión de Proyectos para conocer sus expectativas con respecto a solución a desarrollar, así como profundizar en los aspectos específicos de la Suite Xedro GESPRO a tener en cuenta para el desarrollo de la solución propuesta.

La investigación consta de tres capítulos:

**Capítulo 1: Fundamentación teórica,** en este capítulo se aborda el marco teórico que constituye la base fundamental de la presente investigación. Además se explican los conceptos y las tecnologías utilizadas en el desarrollo de la solución.

**Capítulo 2: Propuesta de solución,** se determina la propuesta de solución a desarrollar, además se realiza el proceso de captura de requisitos, se describe y muestra el diagrama de clases y el de componentes.

**Capítulo 3: Implementación y prueba,** se realiza todo lo relacionado con los flujos de trabajo de implementación y prueba. Se muestra además el diagrama de despliegue y se realiza la validación de los requisitos y funcionalidades del módulo a través de la realización de pruebas funcionales para garantizar la calidad del software.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se profundiza en los conceptos más importantes referentes al tema y sus principales relaciones. Se expone el estudio del estado del arte realizado, estableciendo una comparación entre las principales herramientas de gestión de proyectos con la Suite Xedro GESPRO, en cuanto a sus potencialidades relacionadas con la planificación, la gestión de recursos, software libre; entre otras. Al mismo tiempo se seleccionan las herramientas, tecnologías y metodología de desarrollo de software a utilizar en la implementación de la versión 15.05 del módulo de RRHH para mejorar la planificación y el análisis de los RRHH.

### 1.1. Conceptos asociados

A continuación se mencionan los conceptos relacionados con el objeto de estudio identificado en la investigación.

#### 1.1.1. Recurso

Según el Diccionario de la Lengua Española un “recurso” es una fuente o suministro del cual se produce beneficio. Ayuda o medio al que se recurre para conseguir un fin o satisfacer una necesidad. Es cualquier entidad que contribuye en la realización de actividades del proyecto (3).

#### 1.1.2. Recurso Humano

Los RRHH los constituyen las personas y sus energías, el motor fundamental para el aprovechamiento de esas energías lo constituye la motivación (4). Según Chiavenato son las personas que ingresan, permanecen y participan en la organización, en cualquier nivel jerárquico o tarea (5). De igual forma las personas con las que cuenta una organización (con o sin fines de lucro, y cualquier tipo de asociación) para desarrollar y ejecutar de manera correcta las acciones, actividades, labores y tareas que deben realizarse.

#### 1.1.3. Gestión

La palabra “gestión” proviene del latín “gestio”. Este término hace la referencia a la administración de recursos, sea dentro de una institución estatal o privada, para alcanzar los objetivos propuestos por la misma. Para ello uno o más individuos dirigen los proyectos laborales de otras personas para poder mejorar los resultados, que de otra manera no podrían ser obtenidos (6).

La gestión es el acto de dirigir y controlar a un grupo de personas, con el propósito de coordinar y armonizar el grupo para permitirle lograr un objetivo más allá del alcance del esfuerzo individual. Entre las habilidades de gestión utilizadas por el jefe del proyecto, se incluyen:

- Las habilidades de presentación.
- La negociación.
- Las habilidades de redacción (7).

#### 1.1.4. Gestión de Recursos Humanos

La Gestión de RRHH es el proceso administrativo aplicado al incremento y preservación del esfuerzo, las prácticas, la salud, los conocimientos, las habilidades, etc., de los miembros de la estructura, en beneficio de un sujeto, de la propia organización y del país en general. De igual manera, se puede decir que, realizar el proceso de auxiliar a los empleados a alcanzar un nivel de desempeño y una calidad de conducta personal y social que cubra sus necesidades y expectativas personales (8).

Se refiere a la planeación, organización, desarrollo y coordinación, del talento humano de una organización, así como también al buen control de técnicas, capaces de promover el desempeño eficiente del personal, a la vez que la organización representa el medio que permite a las personas que colaboran en ella alcanzar los objetivos individuales relacionados directa o indirectamente con el trabajo (5) (7).

La Gestión de los RRHH de un proyecto, incluye los procesos que organizan, gestionan y conducen el equipo. El equipo del proyecto está conformado por aquellas personas a las que se les han asignado roles y responsabilidades para completar los objetivos del proyecto (7).

Panorama general de los procesos de Gestión de los RRHH:

- ✓ **Planificación:** Es el proceso por el cual se identifican y documentan los roles dentro de un proyecto, las responsabilidades, las habilidades requeridas y las relaciones de comunicación, y se crea el plan para la dirección de personal.
- ✓ **Adquisición:** Es el proceso por el cual se confirman los RRHH disponibles y se forma el equipo necesario para completar las asignaciones del proyecto.
- ✓ **Desarrollo:** Es el proceso que consiste en mejorar las competencias, la interacción de los miembros del equipo y el ambiente general del equipo para lograr un mejor desempeño del proyecto.

- ✓ **Gestión:** Es el proceso que consiste en dar seguimiento al desempeño de los miembros del equipo, proporcionar retroalimentación, resolver problemas y gestionar cambios a fin de optimizar el desempeño del proyecto (7).

#### 1.1.5. Gestión de Proyectos

La **gestión de proyectos** es la disciplina del planeamiento, la organización, la motivación, y el control de los recursos con el propósito de alcanzar uno o varios objetivos (9).

El *Project Management Institute* (PMI) se refiere a la gestión de proyectos como: la aplicación de conocimiento, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del mismo. Se aplican e integran los procesos de dirección de proyectos: inicio, planificación, ejecución, seguimiento y control, cierre y soporte. En este proceso el director del proyecto es la persona responsable de alcanzar los objetivos del proyecto (7).

#### 1.1.6. Planificación

Es el proceso metódico diseñado para obtener un objetivo determinado. En el sentido más universal, implica tener uno o varios objetivos a realizar junto con las acciones requeridas para concluirse exitosamente. Es un proceso de toma de decisiones para alcanzar un futuro deseado, teniendo en cuenta la situación actual y los factores internos y externos que pueden influir en el logro de los objetivos. Va de lo más simple a lo complejo, dependiendo del medio a aplicarse. La **planificación** consiste en determinar **qué** se debe hacer, **cómo** se debe hacerse, **quién** es el responsable de que se haga y **porqué**. (7)

#### 1.1.7. Sobrecarga de recursos

La sobrecarga o sobreasignación de recursos se produce cuando un recurso tiene asignadas más tareas de las que puede llevar a cabo dentro de su periodo laborable, tanto, días que le correspondan trabajar como el horario que tenga asignado (10) (11).

#### 1.1.8. Fondo de tiempo

Es la cantidad de horas disponibles que un recurso humano dispone a la semana para realizar las tareas asignadas dentro de su periodo laborable.

### 1.2. Estudio comparativo de herramientas con soluciones similares

Gran parte del éxito de un proyecto coincide con que todas las etapas y procedimientos fluyan organizadamente, se conozcan claramente las capacidades de cada organización y se automaticen los procesos (12). Dado el incremento constante de la complejidad para

gestionar esta información, se hace imprescindible el empleo de herramientas que automaticen en mayor medida los procesos relacionados con la gestión de proyectos.

Por consiguiente, se deben encontrar propiedades inherentes en cada herramienta que conlleve al desenvolvimiento natural de dichos procesos. Estas propiedades han de estar orientadas por tanto a la gestión automática de los datos relacionados con los procesos de la gestión de proyectos.

### **1.2.1. Open Project**

*Open Project* es un software de gestión de proyectos, basado en la web libre y de código abierto, basado en *Ruby on Rails*, es multiplataforma. Su licencia es bajo los términos de GNU General Public License (GPL) versión 3. Proporciona funciones de colaboración en todo el ciclo de vida del proyecto, ejemplo, la duración del proyecto, emisión y control de errores, gestión de documentos, gestión de versiones y mucho más. Soporta una variedad de funcionalidades, como la gestión de reuniones, metodologías Scrum, reportes de tiempo y costo (13).

Presenta un módulo de gestión de RRHH el cual permite:

- ✓ Asignar recursos humanos a tareas.
- ✓ Crear dependencias entre tareas.
- ✓ Ver recursos humanos disponibles.
- ✓ Información de recursos humanos centralizada.
- ✓ Manejo de ausencias.
- ✓ Actividades realizadas por los trabajadores y sus habilidades profesionales.

### **1.2.2. Microsoft Project (MSP)**

MSP es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar carga de trabajo de un recurso. No es multiplataforma y no es código abierto (14). Entre sus principales funcionalidades están: ruta crítica, diagrama de Gantt, cálculo de costos y control de proyecto aplicando procedimientos descritos en el *Project Management Body of Knowledge* (PMBOK).

Entre las características relacionadas con la Gestión de Recursos se encuentran:

- ✓ Asignar recursos humanos a tareas.
- ✓ Ver recursos humanos disponibles.

- ✓ Seguir costos.
- ✓ Seguir tiempo.
- ✓ Identificar y resolver la sobrecarga de recursos humanos (14).

### 1.2.3. DotProject

DotProject es una herramienta orientada a la gestión de proyectos basada en la web, soporta varios lenguajes y es código abierto (15). Está desarrollado sobre PHP y usa MySQL como base de datos, es multiplataforma. Entre sus principales módulos están Empresas, Proyecto, Tareas, Usuario y Administración.

Entre las características relacionadas con la Gestión de Recursos se encuentran:

- ✓ Asignar recursos no humanos a un proyecto.
- ✓ Directorio de recursos
- ✓ Seguir costos.
- ✓ Seguir tiempo (15).

### 1.2.4. Collabtive

Collabtive es un software de gestión de proyectos basado en la web, de código abierto. Desarrollada en PHP y Javascript por un equipo de profesionales voluntarios. Destinadas a pequeñas y medianas empresas (16). Ofrece servicios comerciales de instalación y configuración (17). Es fácil de configurar, de usar y entre las principales características se encuentran: es multilinguaje, permite personalizarlo, servicio de mensajería integrado y es simple e intuitivo.

Entre las características relacionadas con la Gestión de Recursos se encuentran:

- ✓ Seguir costos.
- ✓ Asignar recursos.
- ✓ Directorio de recursos.
- ✓ Seguir tiempo. (16) (17).

### 1.2.5. JIRA

JIRA es una herramienta de gestión de proyectos que ayuda a los equipos a construir mejores software. Se sitúa en el centro de su proceso de desarrollo, conectando su equipo con el trabajo. Es multiplataforma y no es de código abierto. Entre sus funcionalidades se pueden mencionar: gestión de proyectos, gestión del portafolio de proyectos.

Entre las características relacionadas con la Gestión de Recursos se encuentran:

- ✓ Asignar recursos.

- ✓ Directorio de recursos.
- ✓ Seguir tiempo (18).

#### 1.2.6. TeamWorkPM

TeamworkPM es una solución en línea para el trabajo en equipo y gestión de proyectos que ayuda a los gerentes, empleados y clientes a trabajar juntos de manera más productiva en línea. Es multiplataforma y no es de código abierto. Sus características incluyen listas de tareas, hitos, cuadernos, mensajes, archivos, registro de riesgos y un calendario mundial.

Entre las características relacionadas con la Gestión de Recursos se encuentran:

- ✓ Asignar RRHH a tareas.
- ✓ Seguir costos.
- ✓ Seguir tiempo (19).

#### 1.2.7. Redmine

Redmine es una herramienta programada con el *framework*<sup>1</sup> Ruby on Rails, bajo código abierto con licencia GPL y posee una interfaz fácil de utilizar. Fue diseñado con una arquitectura que permite adjuntar extensiones con facilidad, además permite tener un mayor control sobre el equipo de trabajo de cada uno de los proyectos que gestiona y facilita el seguimiento de las tareas pues posee elementos que agilizan este trabajo (20).

Entre las características relacionadas con la Gestión de Recursos se encuentran:

- ✓ Asignar recursos.
- ✓ Directorio de recursos.
- ✓ Seguir tiempo (20).

#### 1.2.8. Suite Xedro GESPRO

Suite Xedro GESPRO es un ecosistema para el desarrollo y la innovación en Gestión de Proyectos. Es desarrollado por el Departamento de Gestión de Proyectos de la UCI. Basado en Redmine (20) y otras herramientas libres construidas en la propia organización, el sistema está alineado con los estándares propuestos por el PMI (7) y el *Software Engineering Institute* (SEI) (21).

---

<sup>1</sup> **Framework:** es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto (64).

Esta suite de gestión de proyectos se presenta como un modelo de negocios basado en servicios, donde combina el uso de una solución informática para la gestión de proyectos y un sistema de formación especializada en gestión de proyectos. Posibilita la informatización de la gestión de proyectos en las organizaciones y la mejora continua de sus procesos de planificación, seguimiento y control (22) (23).

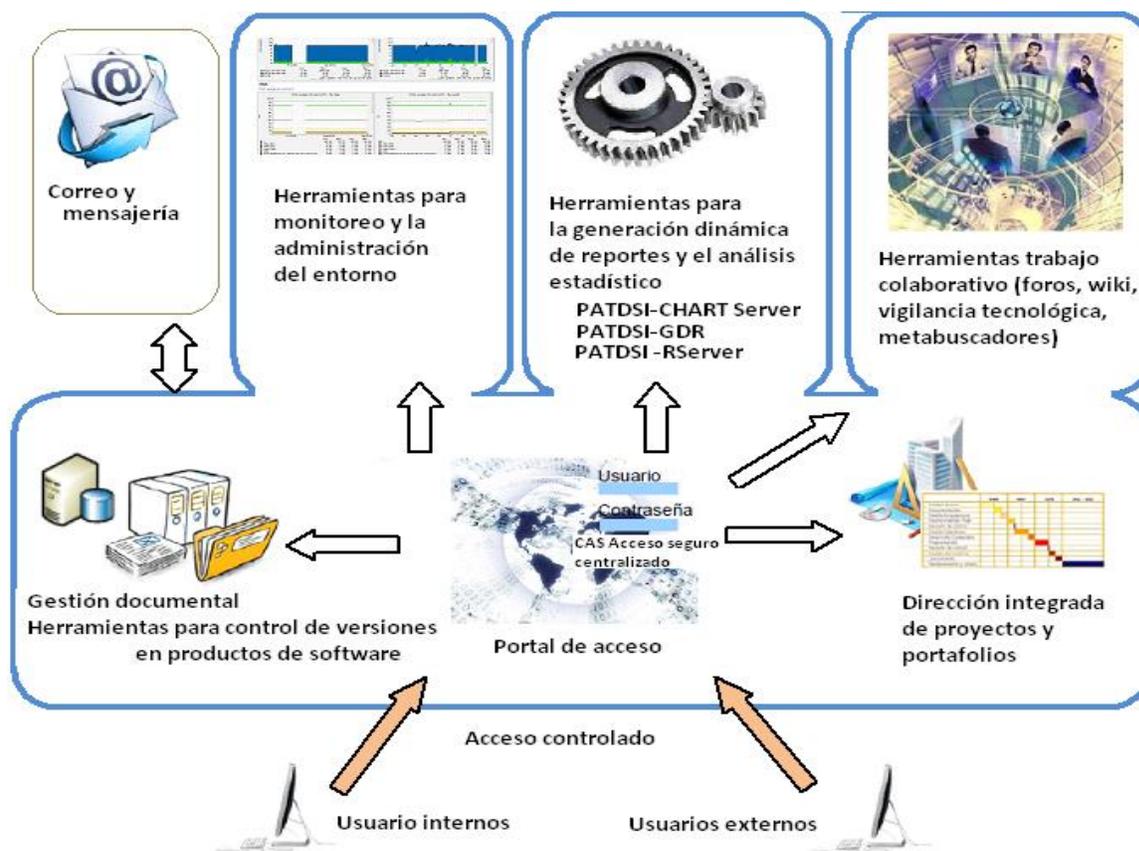
La interacción con la herramienta es a través de la web, por lo que es necesario tener instalado algún navegador web. La solución informática está completamente basada en tecnologías libres y se comercializa bajo licencia GNU-GPL2. El núcleo de la misma es la plataforma Xedro GESPRO 13.05 que es un producto registrado en el Centro Nacional de Derecho de Autor de Cuba, No Registro CENDA 1540-2010 (24).

Ventajas del uso de Xedro GESPRO:

- ✓ Permite el acceso seguro, concurrente y controlado de todo el personal.
- ✓ Aplica algoritmos de *soft computing*<sup>2</sup>.
- ✓ Consulta información para la toma de decisiones.
- ✓ Permite el control y seguimiento de los proyectos.
- ✓ Registro de los resultados de los proyectos en la base de datos de proyectos terminados.
- ✓ Habilita una capa de servicios que garantice la interoperabilidad basada en estándares libres de la suite de gestión de proyectos con otros sistemas de información existentes.
- ✓ Integración con múltiples herramientas (23).
- ✓ Dominado y desarrollado en Cuba (seguridad y soberanía tecnológica).
- ✓ Aplicación web, con un punto de acceso único (ver **Figura 1**).

---

<sup>2</sup>**Soft computing**: es un enfoque emergente de la computación la cual persigue la destacable habilidad de la mente humana para razonar y aprender en un ambiente de incertidumbre e imprecisión (63).



**Figura 1** Sistema de información transaccional Xedro GESPRO.

### 1.2.9. Resultado del estudio de las herramientas

La sociedad cubana se encuentra inmersa en un proceso de migración a sistemas libres de código abierto que permitan alcanzar soberanía y seguridad en la información (25). Con el estudio realizado, el autor considera que la herramienta más idónea a utilizar es la Suite Xedro GESPRO debido a que cumple con el requisito de ser software libre y tener facilidad de extensión, permitiendo su adaptación a las particularidades de cada entorno empresarial.

Además, se concluye que estas herramientas a pesar de satisfacer en gran medida las necesidades de sus clientes, no siempre cubren las expectativas de todos por temas relativos a precio, licencia, soporte o insuficiencias en el manejo de ciertos datos relacionados a la gestión de proyectos.

### 1.3. Metodología de desarrollo de software

El proceso de desarrollo de un software es riesgoso y difícil de controlar por lo que es necesario llevar a cabo una metodología que guíe el proceso a fin de obtener un producto

con calidad. Las metodologías de desarrollo de software son decisivas en el éxito o fracaso de un proyecto, surgen con el objetivo de minimizar el tiempo de desarrollo, hacen muy eficiente y reproducible el camino para obtener resultados confiables. Por ello es de suma importancia establecer la adecuada para así minimizar riesgos e incrementar las posibilidades de éxito en el desarrollo de los productos. (26).

### 1.3.1. Metodologías ágiles

Las metodologías ágiles surgen de la necesidad de proporcionar respuestas rápidas y adaptables al cambio (27). Es un marco metodológico de trabajo que plantea permitir mejorar la eficiencia en la producción y calidad de los productos finales, tener la capacidad de respuesta al cambio y brindar mayor satisfacción al cliente, a través de la entrega temprana y la retroalimentación continua durante el proceso de desarrollo de software.

Al utilizar procesos ágiles para la gestión de proyectos, se perciben beneficios para el cliente:

- ✓ Flexibilidad en el proceso y definiciones del producto.
- ✓ Retroalimentación continua con el cliente.
- ✓ Interacción constante.

Entre las metodologías ágiles más destacadas se encuentran:

- ✓ *Extreme Programming (XP)*.
- ✓ Scrum.
- ✓ Crystal Clear.
- ✓ *Adaptive Software Development (ASD)*.
- ✓ *Lean Software Development (LSD)*.
- ✓ *Dynamic Systems Development Method (DSDM)*.
- ✓ *Feature Driven Development (FDD)*.
- ✓ Kanban
- ✓ *Open Unified Process (OpenUP) (28)*.

Las metodologías ágiles se ajustan al desarrollo de proyectos con requisitos poco definidos o cambiantes. Estas se aplican a equipos de tamaño pequeño que resuelven problemas concretos, lo que no está implacable con su aplicación en el desarrollo de grandes productos, ya que una modularización de los mismos es fundamental para alcanzar el éxito. Dividir el trabajo en módulos minimiza los fallos y el coste.

Es por ello que se seleccionan las metodologías ágiles para el desarrollo de la presente investigación, además la Suite Xedro GESPRO basa su desarrollo en la metodología ágil Scrum.

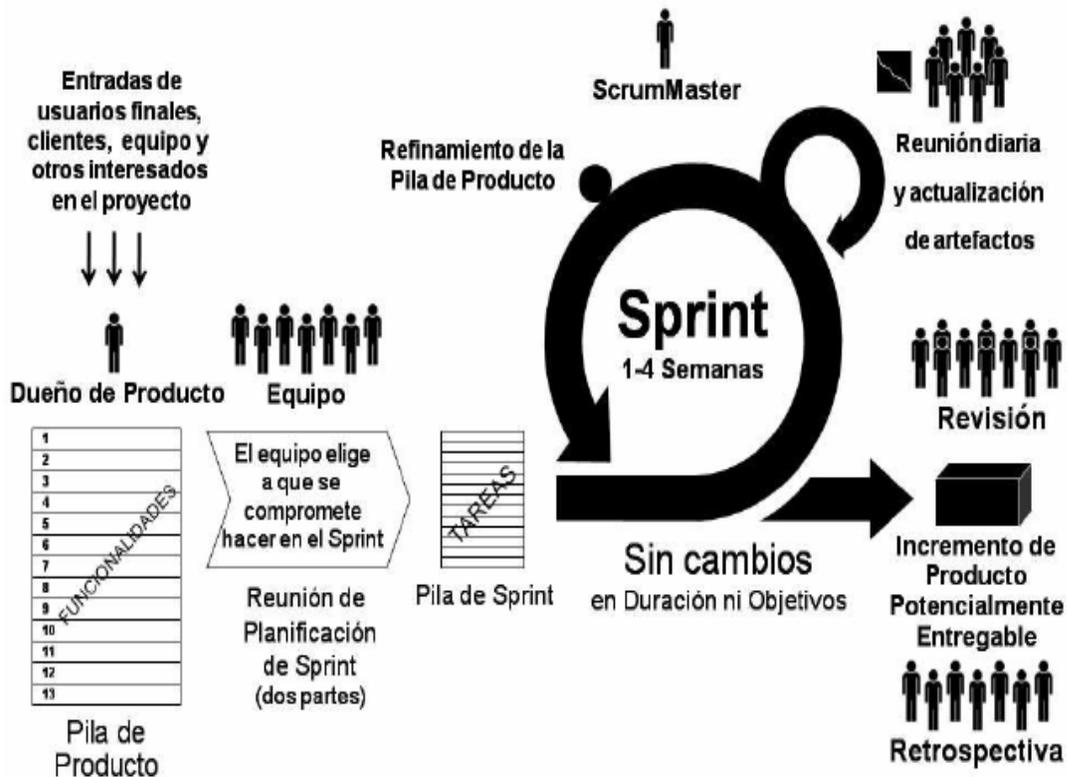
### 1.3.2. Scrum

Scrum es un proceso para la gestión y control del producto que trata de eliminar la complejidad en estas áreas para centrarse en la construcción de software que satisfaga las necesidades del negocio (29). Es simple y escalable, ya que no establece prácticas de ingeniería del software sino que se aplica o combina, fácilmente, con otras prácticas ingenieriles, metodologías de desarrollo o estándares ya existentes en la organización (30).

- ✓ Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos años. Surge como metodología para el desarrollo de productos tecnológicos, pero en la actualidad se emplea en contextos que trabajan con requisitos inconsistentes y que requieren rapidez y flexibilidad; situaciones usuales en determinados sistema de software (31) (32).

Sus principales características se pueden resumir en dos:

- ✓ El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración máxima de 30. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente.
- ✓ La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca una reunión para determinar el trabajo que se va a realizar, otra al final para evaluar el resultado, y revisiones diarias que realiza el equipo en su auto-gestión (ver **Figura 2**) (28).



**Figura 2** Representación del proceso de desarrollo de la metodología Scrum.

Ventajas de Scrum con respecto a otras metodologías ágiles:

- ✓ Cumplimiento de expectativas.
- ✓ Flexibilidad a cambios.
- ✓ Reducción del tiempo.
- ✓ Mayor calidad del software.
- ✓ Máxima productividad.
- ✓ Maximiza el retorno de la inversión.
- ✓ Predicciones de tiempo.
- ✓ Reducción de riesgos (29) (31) (32).

#### 1.4. Herramientas y tecnologías

La solución propuesta será integrada con el módulo de RRHH de la Suite Xedro GESPRO, por lo que se consideró trabajar con las versiones actuales del marco de trabajo existente.

Para dar inicio al desarrollo de la nueva versión del módulo de RRHH, la metodología seleccionada posibilitó realizar previamente un estudio riguroso y detallado de tecnologías, herramientas, lenguaje de modelado, *framework*, servidores web, gestores de bases de

datos y lenguajes de desarrollo definido por el equipo de desarrollo del Departamento de Gestión de Proyectos. Se priorizaron las herramientas y tecnologías de código abierto o pertenecientes al software libre, en las cuales el desarrollador posee experiencias en su uso y que cuenten con una amplia documentación y comunidad de usuarios, aspectos importantes en la retroalimentación y rectificación de errores.

Para el desarrollo de la solución propuesta se utilizará:

#### 1.4.1. Ubuntu

GNU es un acrónimo recursivo que se traduce como "GNU No es Unix". Linux fue adoptado como núcleo del sistema GNU, junto a él y otros muchos programas, constituyen el actual sistema operativo GNU/Linux. Debian es un sistema operativo libre que utiliza el núcleo de Linux pero que la mayor parte de sus herramientas básicas provienen del proyecto GNU.

Ubuntu es un sistema operativo basado en Debian GNU/Linux, y predominantemente, enfocado en la facilidad de uso e instalación, la libertad de los usuarios, y los lanzamientos. El nombre proviene del concepto africano Ubuntu, que significa "humanidad hacia otros" o "yo soy porque nosotros somos". Al igual que otras distribuciones se compone de múltiples paquetes de aplicaciones normalmente distribuidos bajo una licencia libre o de código abierto (33).

La filosofía de Ubuntu se basa en los siguientes principios:

- ✓ El usuario debe tener la libertad de descargar, ejecutar, copiar, distribuir, estudiar, compartir, cambiar y mejorar su software para cualquier propósito, sin tener que pagar derechos de licencia.
- ✓ Debe ser capaz de utilizar su software en el idioma de su elección.
- ✓ Debe ser capaz de utilizar todo el software independientemente de su discapacidad (34).

#### 1.4.2. Herramienta CASE

Las herramientas CASE (siglas de *Computer Aided Software Engineering*), son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas. Su evolución se precede del desarrollo progresivo de la Ingeniería de Software como disciplina. El propósito de una herramienta CASE es dar soporte automatizado para la aplicación de todas o algunas técnicas usadas por una o varias metodologías (35). Las principales herramientas CASE son: *Visual Paradigm, Rational Rose, Umbrello* y *Software Modeling*.

Estas herramientas favorecen de manera directa en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores.

Para el modelado de la propuesta de solución se hará uso de la herramienta Visual Paradigm for UML (VP-UML) en su versión 8.0. Dicha herramienta soporta los estándares más importantes de la industria, tales como el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés), la Notación para el Modelado de Procesos de Negocio (BPMN, por sus siglas en inglés), y otras especificaciones definidas por el grupo de estandarización OMG (del inglés, *Object Management Group*). Además, ofrece un completo conjunto de herramientas que facilitan a los desarrolladores la captura de requisitos, planificación de software, planificación de controles, el modelado de clases, modelado de datos, entre otros (36).

VP-UML es multiplataforma, se encuentra disponible en varias versiones: Enterprise, Professional, Standard, Modeler, Personal y Community. Posee capacidades de ingeniería directa e inversa, y es una herramienta que soporta el ciclo de vida de un software.

### 1.4.3. Lenguaje de modelado

UML es el lenguaje de modelado más conocido y utilizado en la actualidad, entre sus funciones se define como un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software (36). Ofrece un estándar para describir el modelo de un sistema y cuenta con una gran variedad de propiedades entre las que se destacan:

- ✓ Modelar complejas estructuras.
- ✓ Es el más usado a nivel mundial.
- ✓ Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clases, componentes y nodos.
- ✓ Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas (37).

UML se ha convertido ya en una de las mejores herramientas para el diseño y desarrollo de software fiable, eficiente y de calidad. Permite modelar sistemas de información, y su objetivo es lograr modelos que, además de describir con cierto grado de formalismo tales sistemas, puedan ser entendidos por los clientes o usuarios de aquello que se modela.

#### 1.4.4. Lenguajes de desarrollo del lado del cliente

##### HTML5

HTML5 (HyperText Markup Language, versión 5) establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Entre los nuevos atributos que se destacan se encuentran:

**Tabla 1** Nuevos atributos del HTML5. Fuente: (38).

Etiquetas	Atributos
<article>	Atributos globales
<audio>	autobuffer   autoplay   controls   loop   src
<canvas>	height   width
<command>	checked   default   disabled   hidden   icon   label   radiogroup   type
<dialog>	Atributos globales
<embed>	height   src   type   width
<figure>	Atributos globales
<footer>	Atributos globales
<header>	Atributos globales
<hgroup>	Atributos globales
<meter>	high   low   max   min   optimum   value
<output>	form
<progress>	max   value
<ruby>	cite
<section>	cite
<nav>	Atributos globales
<aside>	Atributos globales
<svg>	Atributos globales
<source>	media   src   type
<time>	datetime   pubdate
<video>	src   poster   autobuffer   autoplay   loop   controls   width   height

Presenta nuevas características, tales como:

- ✓ Incorpora etiquetas (*canvas*<sup>3</sup> 2D y 3D, audio, video) con codecs (codificador - decodificador) para mostrar los contenidos multimedia.
- ✓ Etiquetas para manejar grandes conjuntos de datos. Permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido al cliente.
- ✓ Mejoras en los formularios. Nuevos tipos de datos y facilidades para validar el contenido sin Javascript.
- ✓ Añade etiquetas para manejar la Web Semántica (Web 3.0) (38).

<sup>3</sup> **Canvas**: un canvas es un lienzo de mapa de bits dependiente de la resolución de pantalla.

### CSS3

CSS del inglés *Cascading Style Sheets* es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML<sup>4</sup> (39). El uso del CSS es para separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML.

Para la propuesta de solución se usará el CSS3 ya que en diferencia a versiones anteriores, está dividida en varios documentos separados, llamados "módulos". Cada módulo añade nuevas funcionalidades a las definidas en CSS2, de manera que se preservan las anteriores para mantener la compatibilidad (40).

#### 1.4.5. Lenguaje de desarrollo del lado del servidor

Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos. Creado por el programador japonés Yukihiro "Matz" Matsumoto. Soporta múltiples paradigmas de programación, incluyendo funcional, orientado a objetos e imperativa. Cuenta además con un tipo dinámico del sistema y automáticamente la gestión de memoria, por lo que es parecido en varios aspectos a Smalltalk, Python, Perl, Lisp, Dylan, Pike, y CLU. Es de código abierto bajo licencia BSD (del inglés, *Berkeley Software Distribution*).

Entre sus principales características:

- ✓ Orientado a objetos.
- ✓ Cuatro niveles de ámbito de variable: global, clase, instancia y local.
- ✓ Carga dinámica de DLL/bibliotecas compartidas en la mayoría de las plataformas.
- ✓ Ruby es fácilmente portable: se desarrolla mayoritariamente en GNU/Linux, pero corre en varios tipos de UNIX, Mac OS X, Windows 95/98/Me/NT/2000/XP/7/8, DOS, BeOS, OS/2, etc. (41).
- ✓ Tiene manejo de hilos independiente del sistema operativo. De esta forma, tiene soporte multi-hilos en todas las plataformas en las que corre Ruby, sin importar si el sistema operativo lo soporta o no.

#### 1.4.6. Framework de desarrollo del lado del servidor

El *framework* Ruby on Rails también conocido como RoR o Rails es sobre el cual está desarrollada la Suite Xedro GESPRO. RoR es de código abierto y está escrito en el lenguaje de programación Ruby. Trata de combinar la simplicidad con la posibilidad de desarrollar

---

<sup>4</sup> **XML**: Lenguaje de Marcado Extensible, de sus siglas en inglés de *eXtensible Markup Language*

aplicaciones del mundo real escribiendo menos código que con otros *frameworks* y con un mínimo de configuración (42).

Sigue dos principios básicos:

- ✓ Convención sobre la configuración: Con Rails, en lugar de archivos de configuración se utilizan una serie de convenciones. Es decir que no es necesario configurar la aplicación en un archivo XML (por ejemplo), basta con utilizar una convención que sustituya la configuración.
- ✓ Menos software: Se refiere a que se escriben menos líneas de código, lo que trae consigo más rapidez y menos errores en la implementación.

#### 1.4.7. Gestor de Base de Datos

Los Sistemas de Gestión de Bases de Datos (SGBD) son un tipo de software específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Existen muchos SGBD pero en su gran mayoría son sistemas propietarios entre los que se destacan: *MySQL*, *Advantage Database*, *FileMaker*, etc. También se encuentran SGBD disponibles como sistemas libres como son: *PostgreSQL*, *FileMaker*, *MySQL*, etc.

Como gestor de base de datos para la solución propuesta se utilizará *PostgreSQL* en su versión 9.1. *PostgreSQL* es un SGBD Relacional orientado a objetos y libre, publicado bajo la licencia BSD. Como muchos otros proyectos de software libre, el desarrollo de *PostgreSQL* es manejado por una comunidad de desarrolladores y organizaciones comerciales, dicha comunidad es denominada el *PostgreSQL Global Development Group* (PGDG) (43).

Algunas de las características de este sistema son:

- ✓ Alta concurrencia.
- ✓ Alta variedad de tipos nativos.
- ✓ Claves Ajenas.
- ✓ Disparadores.
- ✓ Vistas.
- ✓ Tipos de datos y operaciones geométricas.
- ✓ Integridad transaccional.
- ✓ Herencia de tablas.
- ✓ Soporte para transacciones distribuidas.
- ✓ Funciones (44) (45).

**PgAdmin3** es una aplicación gráfica para el gestor de bases de datos PostgreSQL con licencia Open Source, escrita en C++ usando la biblioteca gráfica multiplataforma wxWidgets. Esto permite que se pueda usar sobre las plataformas Linux, FreeBSD, Solaris, Mac OS X, Windows. Capaz de manejar versiones de PostgreSQL a partir de la 7.3 ejecutándose sobre cualquier plataforma, así como versiones comerciales como *Pervasive Postgres*, *EnterpriseDB*, *Mammoth Replicator* y *SRA PowerGres*.

La interfaz gráfica es compatible con todas las características de *PostgreSQL* y facilita la administración. La conexión con el servidor se puede realizar mediante TCP/IP o *Unix Domain Sockets* y puede ser cifrado mediante SSL por seguridad. No se requieren controladores adicionales para comunicarse con la base de datos del servidor e incorpora funcionalidades para realizar consultas, examinar su ejecución y trabajar con los datos (46).

#### 1.4.8. Servidor web

Un servidor web es un programa informático para procesar las aplicaciones del lado del servidor mediante conexiones que generan una respuesta desde las aplicaciones del lado del cliente. El servidor web seleccionado para utilizar en la propuesta de solución es Apache en su versión 2.2.22 (47).

Apache es el servidor web multiplataforma de código abierto, hecho por excelencia, su robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Permite generar informes de errores HTTP<sup>5</sup> y gestionar recursos para procesos hijos. El servidor Apache presenta entre otras características: mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor web Apache, existen gran cantidad de módulos Apache disponibles para su utilización. Tiene una buena compatibilidad con el lenguaje PHP compartiendo muchas de sus características.

Principales características de Apache:

- ✓ **Fiabilidad:** Alrededor del 90% de los servidores con más alta disponibilidad funcionan con Apache.
- ✓ **Gratuidad:** Apache es totalmente gratuito, y se distribuye bajo la licencia Apache Software License, favoreciendo la modificación del código.

---

<sup>5</sup> **HTTP:** siglas de Protocolo de Transferencia de Hipertexto (del inglés, *Hypertext Transfer Protocol*).

- ✓ Extensibilidad: Se pueden añadir módulos para ampliar las amplias capacidades de Apache. Existe una amplia variedad de módulos, que permiten generar contenido dinámico con PHP, Java, Perl, Ruby, entre otros, además de monitorizar el rendimiento del servidor. Estos módulos pueden ser creados por cualquier persona con conocimientos de programación (47).

#### 1.4.9. Entorno de desarrollo integrado

##### NetBeans v6.9

Un entorno de desarrollo integrado, llamado también IDE (siglas en inglés de *Integrated Development Environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Para la propuesta de solución se seleccionó el IDE NetBeans en su versión 6.9. NetBeans es un entorno de desarrollo integrado, multiplataforma, de código abierto, permite a los desarrolladores crear rápidamente aplicaciones web, empresa, móviles y de escritorio con el uso de plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y *Ruby on Rails*, Groovy y Grails, y C/C++. Es gratuito, tiene una gran comunidad de usuarios y desarrolladores en todo el mundo, proporciona herramientas de análisis estático para identificar y solucionar problemas comunes en código Java; además el IDE tiene editores y herramientas para XML, HTML, PHP, Ruby y *Ruby on Rails*, Groovy y Grails, Javadoc, JavaScript y JSP (48).

El IDE NetBeans 6.9 ofrece un rendimiento significativamente mejorado y mayor experiencia de codificación. La base en la que se sustenta su elección es que permite desarrollar aplicaciones utilizando el *framework Ruby on Rails* y ejecutar los comandos del mismo directamente desde la interfaz del IDE.

#### 1.5. Conclusiones parciales

A partir del análisis de las cuestiones expresadas en este capítulo es posible concluir que:

Se selecciona como herramienta de gestión de proyectos la Suite Xedro GESPRO, por las facilidades y potencialidades que brinda esta herramienta a partir de la integración de varios módulos que facilitan la gestión de los proyectos.

Las herramientas de gestión de proyectos estudiadas no integran las distintas funcionalidades y facilidades para mejorar el proceso de análisis y planificación de recursos humanos, teniendo en cuenta la sobrecarga de los recursos humanos.

Se selecciona como metodología de desarrollo, la metodología ágil SCRUM, a pesar de que son disímiles las metodologías de desarrollo de software que se utilizan en el mundo para gestionar los proyectos, ya que la Suite Xedro GESPRO posee en su marco de trabajo esta metodología de desarrollo.

A partir del cumplimiento del primer objetivo específico se concluye que no existe un método o procedimiento que facilite la planificación y el análisis en los entornos reales de proyectos de software teniendo en cuenta la sobrecarga de recursos humanos, de ahí la necesidad de desarrollar nuevas funcionalidades para la gestión de recursos humanos en el Módulo de Recursos Humanos de la Suite Xedro GESPRO.

## CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

En el presente capítulo se realiza la descripción de la propuesta de solución de esta investigación, para lo que se desarrollan las actividades de análisis y especificación de requisitos. Además se describe la arquitectura del módulo y se realizan diferentes artefactos necesarios en el proceso de diseño e implementación del mismo.

### 2.1. Propuesta de solución

Una vez identificada que la correcta gestión de los recursos humanos, partiendo de las capacidades que presentan, su disponibilidad y sobrecarga, garantizarán el éxito de la organización y definidas las deficiencias que presenta la Suite Xedro GESPRO en el área de Recursos Humanos, con respecto a la sobrecarga de RRHH, es preciso proponer una solución que incorpore al módulo de RRHH de la suite, un conjunto de funcionalidades que mejoren el proceso de planificación y análisis de los RRHH.

Primeramente se rediseñará la vista de presentación del módulo de RRHH, alineada a los principales procesos del área de conocimiento Gestión de RRHH que propone PMBOK. En un segundo momento se incorporará el índice de rendimiento del recurso humano con respecto a su aprovechamiento (IRHA), a la vista actual del proceso Gestión del módulo de RRHH. En un tercer momento se diseñará el calendario de tareas concurrentes, donde se evidencie la sobrecarga del recurso humano y por último, se añadirá el IRHA al cronograma de tareas, lo cual permitirá conocer la disponibilidad de cada recurso humano en el momento de crear una nueva petición.

A continuación se describe el coeficiente que calcula el índice de aprovechamiento de los recursos humanos:

Índice de Rendimiento del Recurso Humano con respecto a su Aprovechamiento (IRHA):

Relación entre los totales de tiempo planificado (TTP) y disponible (TTD) asignados al recurso humano hasta la fecha del corte.

Interpretación:

$IRHA < 0.75$ , disponible;

$0.75 \leq IRHA \leq 1$ , disponible con riesgo;

$IRHA > 1$ , sobrecargado.

$$IRHA(corte) = TTP/TTD (49).$$

## 2.2. Especificación de requisitos

La escritura de los requisitos se refiere a la tarea de reunir una descripción del producto desde el punto de vista de negocio. Los requisitos obtenidos fueron especificados de manera correcta en la plantilla 0115\_Gespro\_RRHH.Especificación De Requisitos De Software con la cual quedan descritos de manera correcta los requisitos identificados para la versión 15.05 del módulo RRHH de la Suite Xedro GESPRO (para obtener mayor detalle dirigirse al documento adjunto a la investigación).

### 2.2.1. Requisitos funcionales

Los requisitos funcionales describen las capacidades o funciones que el sistema debe cumplir, los servicios que de él se esperan, o los que proveerá proporcionada por uno o más componentes de un sistema.

Luego de un análisis de los requerimientos del cliente, se obtuvieron los siguientes requisitos funcionales:

- RF 1. Diseñar la vista de presentación del módulo de RRHH alineada a los principales procesos que propone la Gestión de RRHH según el PMBOK.
  - RF 1.1 Incluir pestaña para el proceso de Planificación.
  - RF 1.2 Incluir pestaña para el proceso de Adquisición.
  - RF 1.3 Incluir pestaña para el proceso de Desarrollo.
  - RF 1.4 Incluir pestaña para el proceso de Gestión.
- RF 2. Incorporar a la vista actual de Gestión de RRHH el indicador de sobrecarga.
  - RF 2.1 Mostrar listado de RRHH con su índice de sobrecarga.
  - RF 2.2 Colorear el índice de sobrecarga de los RRHH (Verde, Naranja o Rojo).
  - RF 2.3 Ordenar ascendentemente o descendientemente la lista de RRHH según el índice de sobrecarga.
- RF 3. Diseñar calendario de tareas concurrentes donde se evidencie la sobrecarga.
  - RF 3.1 Mostrar listado de tareas concurrentes por personas.
  - RF 3.2 Mostrar por mes y año las tareas.
- RF 4. Rediseñar la vista de RRHH en el cronograma de tareas resaltando el coeficiente de sobrecarga.
  - RF 4.1 Colorear el índice de sobrecarga de los RRHH (Verde, Naranja o Rojo).

RF 4.2 Mostrar mensaje de disponibilidad (Disponible, No disponible, Disponible con riesgo).

### 2.2.2. Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Los requisitos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto (50).

Se definen los siguientes requisitos no funcionales:

*Tabla 2 Requisitos no funcionales.*

Hardware	
<b>RNF1</b>	Servidor Aplicaciones Web: 2GB RAM, 250 GB disco duro.
<b>RNF2</b>	Servidor Base de Datos: 2GB RAM, 250 GB disco duro.
<b>RNF3</b>	PC Cliente: Prestaciones de hardware que permitan la instalación de cualquier Sistema Operativo, conexión de red y un navegador para acceder a la plataforma.
Software	
<b>RNF4</b>	PC Cliente con navegador Mozilla Firefox v13.0 o superior.
<b>RNF5</b>	PC servidor de base de datos: El servidor debe contar con sistema operativo GNU/Linux Ubuntu 12.04 LTS y el SGDB PostgreSQL 9.1.
<b>RNF6</b>	Como servidor de aplicación Apache 2.
<b>RNF7</b>	Lenguaje de programación: Ruby.

<b>Eficiencia</b>	
<b>RNF8</b>	El sistema debe soportar un tiempo de respuesta menor o igual a 5 segundos.
<b>RNF9</b>	El sistema debe soportar una conexión simultánea de más de 3000 usuarios.
<b>Requisitos de apariencia o interfaz externa</b>	
<b>RNF10</b>	La interfaz es sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo.
<b>RNF11</b>	Debe utilizar como idioma principal el español, aunque debe existir traducción al inglés.
<b>RNF12</b>	Los botones expresarán su función ya sea mediante su texto o la imagen que lo acompañe.
<b>Interfaces de hardware</b>	
<b>RNF13</b>	La comunicación entre el servidor de aplicaciones y la base de datos se lleva a través del protocolo de conexión TCP/IP.
<b>RNF14</b>	La comunicación entre el cliente y el servidor de aplicaciones se lleva a través del protocolo HTTPS.
<b>Requisitos Legales, de Derecho de Autor y otros</b>	
<b>RNF15</b>	El sistema debe ser sometido a un análisis legal por parte de los abogados y personal autorizado con vistas a declarar su autenticidad y evitar restricciones legales para su uso y comercialización; así mismo se

	debe proceder a una evaluación y certificación por parte del cliente del producto.
<b>Seguridad</b>	
<b>RNF16</b>	El sistema debe estar disponible las 24 horas los 365 días del año, para garantizar la ejecución de las tareas en el momento requerido.
<b>RNF17</b>	Permite el acceso al sistema a través de una cuenta de usuario única que puede ser local o por autenticación con el LDAP.
<b>RNF18</b>	El acceso por las cuentas de administración a los servidores se realiza a través de una contraseña generada con algoritmo MD5. El listado de las claves de cada servidor se almacena en un archivo comprimido cifrado.
<b>RNF19</b>	Los usuarios creados en la Suite Xedro GESPRO tienen acceso (o permiso) limitado, podrán acceder a un conjunto de funcionalidades según el rol o perfil donde fue clasificado el usuario, que a su vez este rol fue clasificado en un nivel para la toma de decisiones determinado.
<b>Usabilidad</b>	
<b>RNF20</b>	Facilidad de uso por parte de los usuarios: el sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.
<b>RNF21</b>	Especificación de la terminología utilizada: el sistema debe adaptarse al lenguaje y términos utilizados por los usuarios en la rama abordada con vista a una mayor comprensión por parte del cliente de la herramienta de trabajo.

<b>RNF22</b>	Potencialidades de capacitación orientadas a interfaces intuitivas, lo que enaltece la posibilidad de que el usuario aprenda mediante el uso y explotación de la herramienta.
<b>RNF23</b>	El sistema debe presentar una serie de menús tanto laterales como en barra de íconos flotantes que permitan el acceso rápido a la información por parte de los usuarios, lo que aprovecha las potencialidades de estas estructuras.

### 2.3. Diseño del módulo

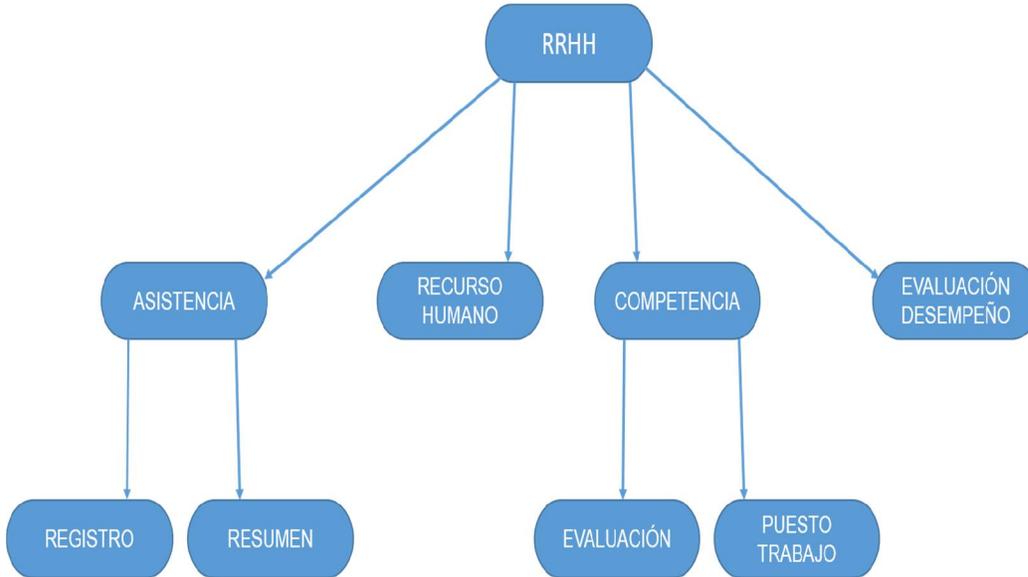
El diseño es una representación de la ingeniería en el desarrollo de aplicaciones. Es la etapa donde se fomentará la calidad de la ingeniería, en la cual se proporcionan las representaciones del software que se pueden evaluar en cuanto a calidad. Es la primera etapa técnica del proceso de ingeniería de software, consiste en un modelo o representación técnica del software que se va a desarrollar. Es la única forma de convertir exactamente los requisitos de un cliente en un producto o sistema finalizado y sirve como fundamento para todos los pasos siguientes del soporte y de la ingeniería del software.

Sin un diseño, se corre el riesgo de construir un sistema inestable, un sistema que fallará cuando se lleven a cabo cambios; un sistema que puede resultar difícil de comprobar; y un sistema cuya calidad no puede evaluarse hasta muy avanzado el proceso, sin tiempo suficiente y con mucho dinero gastado en él (51). En el contexto de la ingeniería de software, el diseño se centra en cuatro áreas importantes de interés: Arquitectura, interfaces, datos y componentes.

A continuación se exponen los fundamentos del módulo de RRHH para la Suite Xedro GESPRO, utilizando como punto de partida el diseño estructural del módulo actual y el nuevo diseño propuesto, la arquitectura del mismo seguido de los patrones de diseño empleados y los diferentes artefactos generados en el proceso de diseño e implementación.

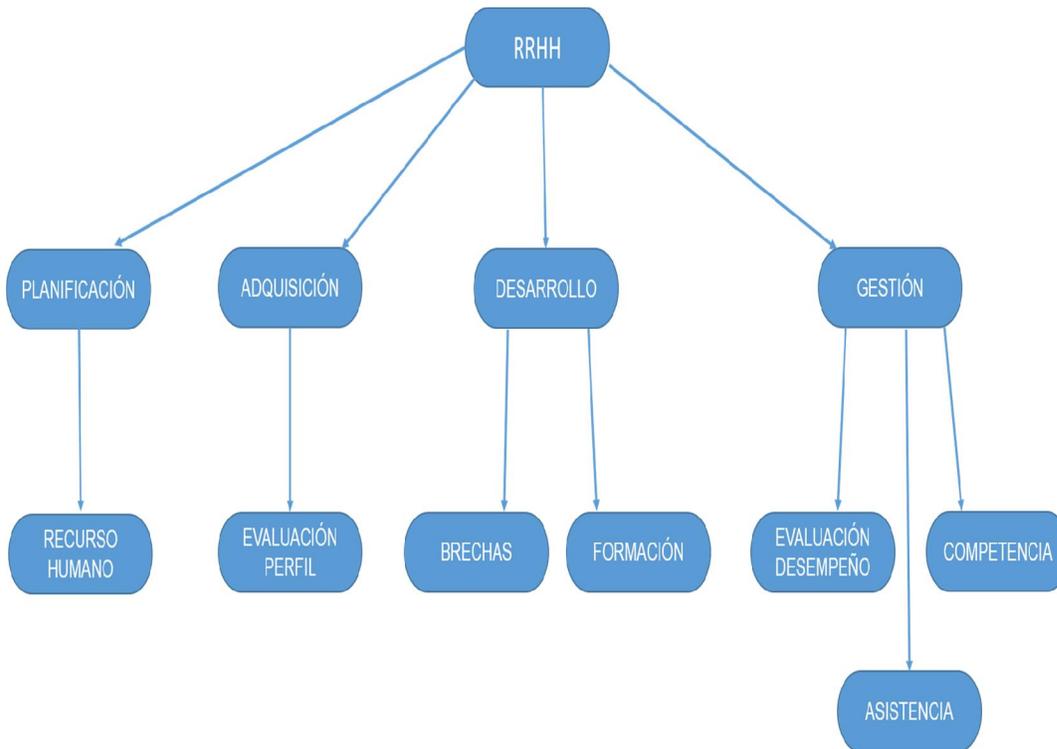
En la **Figura 3** se muestra el diseño de la estructura del módulo de RRHH en la versión 13.05, el cual será analizado y reformado para lograr el objetivo de la presente

investigación.



**Figura 3** Estructura del módulo de RRHH de la Suite Xedro GESPRO v13.05.

Después de un análisis exhaustivo se decide conformar la nueva estructura (ver **Figura 4**) del módulo de RRHH para que esté alineado con los procesos que plantea el PMBOK, pilar fundamental por el cual se rige la Suite Xedro GESPRO.



**Figura 4** Nueva estructura del módulo de RRHH de la Suite Xedro GESPRO.

## 2.4. Patrón de arquitectura

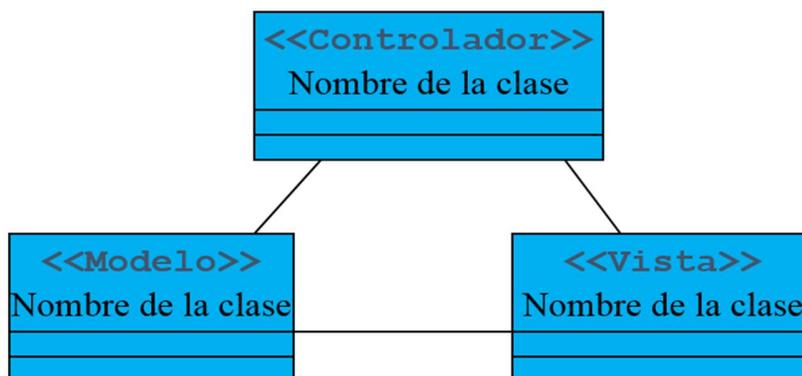
Los patrones de arquitectura son aquellos que expresan un esquema organizativo estructural fundamental para sistemas de software. Describen prácticas para definir la arquitectura del sistema. Proporcionan un conjunto de subsistemas o módulos predefinidos, con reglas y guías para organizar las relaciones entre ellos.

Los patrones arquitectónicos determinan:

- La organización estructural del sistema.
- La selección de elementos estructurales.
- El comportamiento de los componentes.
- Las interfaces entre ellos (52).

### Modelo Vista Controlador

El Modelo Vista Controlador (MVC) es un patrón de diseño de arquitectura que se usa principalmente en aplicaciones que procesan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el desarrollo esté estructurado de manera eficiente, facilitando la programación en diferentes capas y de manera paralela e independiente (50). Busca agrupar los componentes de la aplicación en tres niveles lógicos.



**Figura 5** Niveles lógicos de la arquitectura MVC.

El *framework* RoR sigue el paradigma de esta arquitectura (ver **Figura 6**) que está estructurada de la siguiente forma:

**Modelo:** Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador). RoR define en un directorio llamado “*models*” todas las clases consideradas como modelos, que se utilizan para la comunicación con la base de datos.

**Vista:** Es responsable de generar una interfaz de usuario, normalmente basada en los datos del modelo. En el caso de una aplicación web, la vista es una página HTML con contenido dinámico sobre el cual el usuario puede realizar distintas operaciones. RoR define un conjunto de vistas en un directorio llamado “views” para cada funcionalidad básica, en un sub-directorio con el mismo nombre del controlador al que responden.

**Controlador:** Se encarga de manejar y responder las solicitudes del usuario interactuando de esta manera con el modelo para procesar la información necesaria, o sea, organiza la aplicación recibiendo eventos del exterior. Los controladores administran las acciones de RoR, y se encuentran en un directorio llamado “controllers”.

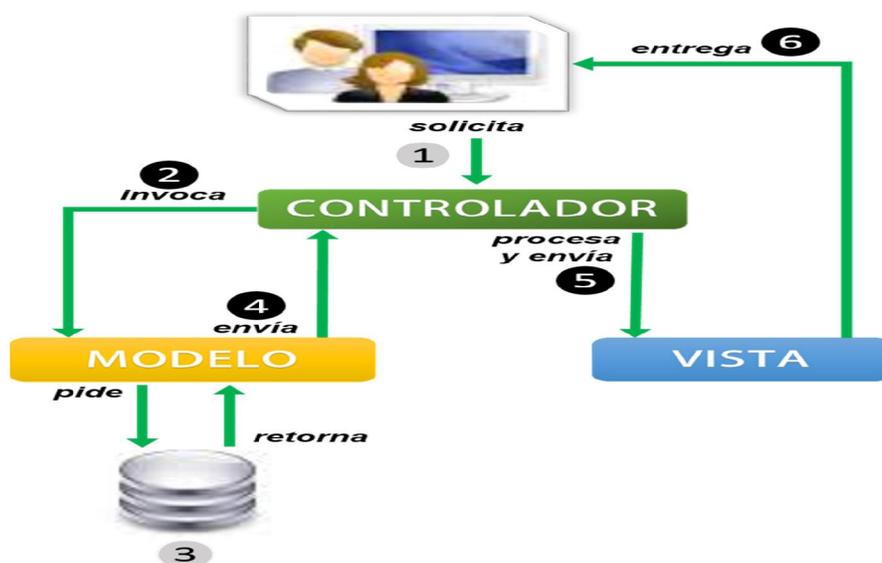


Figura 6 Arquitectura del framework RoR, basado en el patrón MVC.

## 2.5. Patrones de diseño

Los patrones de diseño son descripciones de clases cuyas instancias colaboran entre sí. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema. Representa un esquema o micro-arquitectura que supone una solución a problemas semejantes; una estructura común que tienen aplicaciones similares.

Los patrones de diseño brindan una solución generalmente ya probada y documentada a problemas que se dan durante el proceso de desarrollo de software (53). Emplean un conjunto de buenas prácticas que facilitan el trabajo, definen una estructura de clases que da respuesta a uno o varios problemas en particular y presentan la ventaja de que son fáciles de comprender, además de que no dependen del lenguaje, haciéndolos genéricos.

Lo complejo es cuando se tiene que decidir cuál usar, pues presentan diferentes soluciones, ya sea a través del empleo de uno u otro, o la combinación de varios.

### Patrones GRASP

Los patrones GRASP acrónimo que significa *General Responsibility Assignment Software Patterns* (patrones generales de software para asignar responsabilidades) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (52).

Los empleados para el módulo son los siguientes:

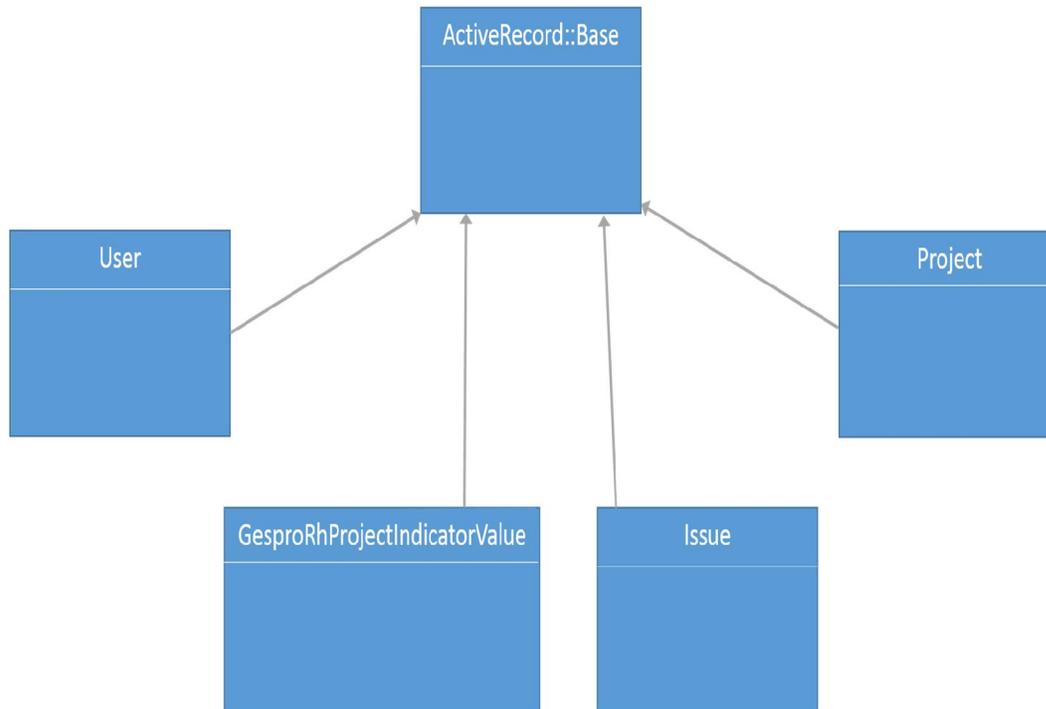
- ✓ **Experto:** se usa más que cualquier otro patrón al asignar responsabilidades en información; es un principio básico que suele utilizarse en el diseño orientado a objetos. Brinda soporte a un bajo acoplamiento y a una alta cohesión, ya que se conserva el encapsulamiento y el comportamiento se distribuye entre las clases que cuentan la información requerida, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento.
- ✓ **Creador:** guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento.
- ✓ **Bajo Acoplamiento:** estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad.
- ✓ **Alta Cohesión:** es un principio que se debe tener presente en todas las decisiones de diseño: es la meta principal que ha de buscarse en todo momento. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas.
- ✓ **Controlador:** este patrón define que clase administra los eventos del sistema. Un controlador es un objeto que no pertenece a la interfaz de usuario, responsable de recibir o manejar un evento del sistema. Su uso se evidencia en las clases controladoras, pues estas tienen la responsabilidad de recibir o manejar los eventos

del sistema. En la aplicación se dividen los eventos del sistema en varias clases controladoras para lograr disminuir el acoplamiento y aumentar la cohesión. Este patrón es utilizado en el módulo de Recursos Humanos en las clases controladoras, ejemplo de ella se tiene *GesproRhProjectIndicatorValuesController*, la misma es la responsable de controlar los eventos del módulo.

## 2.6. Patrón de acceso a datos

El patrón de Acceso a Datos es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos (54). *Active Record* (AR) sigue el estándar de Mapeo Objeto-Relacional (ORM) y se diferencia de los demás porque minimiza la cantidad de configuraciones mediante el uso de un conjunto de convenciones (55) (56).

Cada clase AR representa una tabla de la base de datos cuyos atributos son representados como las propiedades de la clase y una instancia AR representa una fila en esa tabla. Las operaciones CRUD<sup>6</sup> comunes son implementadas como métodos de la clase. Como resultado, se puede acceder a los datos de una manera más orientada a objetos (57).



**Figura 7** Ejemplo del uso de patrón Active Record.

<sup>6</sup> **CRUD**: acrónimo de Crear, Obtener, Actualizar y Borrar (del inglés: *Create, Read, Update and Delete*), es usado para referirse a las funciones básicas en bases de datos o la capa de persistencia en un software.

## 2.7. Modelo Entidad Relación

El Modelo de Entidad Relación es un modelo de datos basado en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones entre estos objetos.

Una entidad caracteriza a un tipo de objeto, real o abstracto, del problema a modelar. Toda entidad tiene existencia propia, es distinguible del resto de las entidades, tiene nombre y posee atributos definidos en un dominio determinado. Una entidad es todo aquello de lo que se desea almacenar información.

Con este modelo se logra representar de manera gráfica la estructura lógica de la base de datos. El módulo de RRHH incorpora una nueva tabla necesaria para dar respuesta a la propuesta de solución. La **Figura 8** muestra la estructura de la base de datos del módulo de RRHH de la Suite Xedro GESPRO.



**Figura 8** Diagrama entidad relación del módulo de RRHH.

**gespro\_rh\_project\_indicator\_values:** en esta tabla se almacenan los indicadores que se calculan para un proyecto y cada uno de sus miembros, además de la evaluación de los

mismos. El atributo *irha* (índice de aprovechamiento) es el utilizado para referirse a la sobrecarga de recursos.

**users:** en esta tabla se guardan parte de los datos de los usuarios que son los RRHH que pertenecen a los proyectos que se gestionan en la Suite Xedro GESPRO.

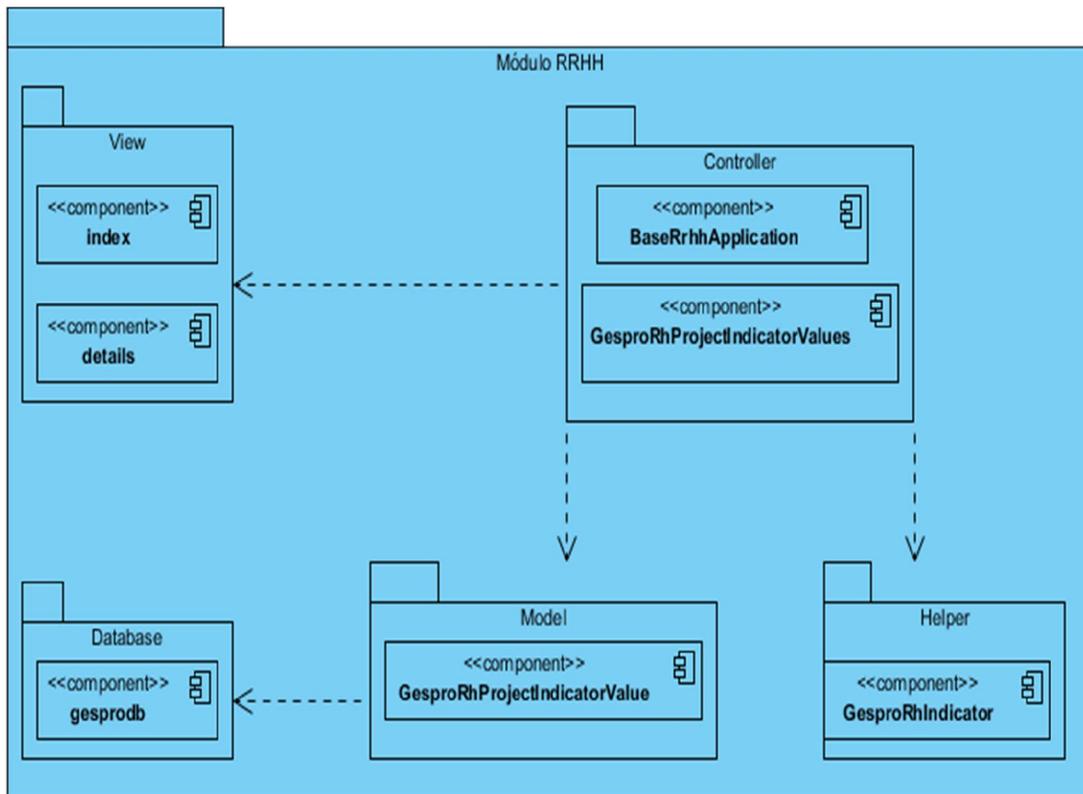
**projects:** en esta tabla se almacena la información correspondiente a cada proyecto gestionado por la Suite Xedro GESPRO.

**members:** esta tabla es el resultado de la relación de mucho a mucho de las tablas *users* y *projects*, en la cual encuentran los miembros de los proyectos.

## 2.8. Diagrama de componentes

El diagrama de componentes permite visualizar con facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. En ellos se muestran los elementos de diseño de un sistema de software y sus dependencias (58). En el diagrama de componentes del módulo de RRHH se agruparon los componentes comunes en paquetes, y se definió la relación entre éstos (ver **Figura 9**).

De esta manera, todos los componentes controladores están dentro del paquete *Controller*, los componentes vistas están dentro del paquete *View* y los componentes modelos están dentro del paquete *Model*. Además se representan componentes externos como las base datos y clases auxiliares utilizadas.



*Figura 9 Diagrama de componentes de los módulos de RRHH.*

## 2.9. Conclusiones parciales

A lo largo de este capítulo se identificaron los requisitos funcionales y no funcionales del módulo de RRHH, permitiendo obtener una descripción en lenguaje natural de las necesidades del cliente. Se propusieron además los patrones de arquitectura y diseño utilizados en el desarrollo del mismo, los cuales constituyen la base para la implementación exitosa, pues describen la estructura y relación de los componentes del módulo.

## CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

En el presente capítulo se documenta la fase de desarrollo con los artefactos que la misma genera según la metodología previamente seleccionada. Se expone el diagrama para el despliegue de la Suite Xedro GESPRO. Se determina la validez de la propuesta desarrollada a través de la aplicación de pruebas de software al módulo y los resultados que estas generan con el propósito de verificar si los requisitos identificados realmente definen una nueva versión del módulo de RRHH.

### 3.1. Implementación del módulo

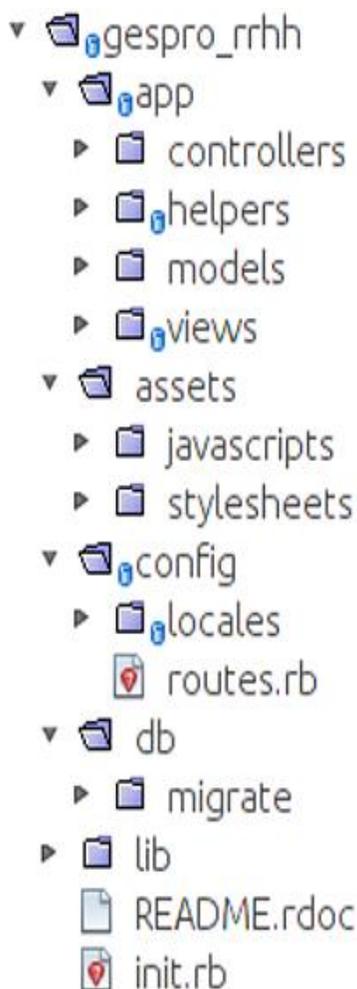
En la fase de implementación en el desarrollo de un producto de software, se ponen en práctica todas las descripciones y arquitecturas propuestas en la fase de análisis y diseño, es el complemento del trabajo de las fases que lo preceden dentro del proceso unificado de desarrollo de software. Ofrece una materialización precisa de los requisitos y en ella se obtienen como resultado componentes de código que se compilan e integran en versiones ejecutables.

#### 3.1.1. Arquitectura del módulo

El módulo de RRHH se encuentra estructurado de la siguiente manera:

- ✓ **app:** Contiene el núcleo del módulo dividido en cuatro subdirectorios *controllers*, *models*, *views* y *helpers*.
  - **controllers:** Contiene las clases controladoras del módulo.
  - **models:** Reúne las clases de modelación de datos almacenados en la base de datos de la aplicación.
  - **views:** Contiene las plantillas de visualización que se llenarán con los datos de la aplicación, las mismas que serán convertidas en HTML y devueltas al navegador del usuario.
  - **helpers:** Contiene las clases auxiliares que proporcionan funcionalidades complementarias.
- ✓ **assets:** Contiene los subdirectorios *stylesheets* y *javascripts*.
  - **stylesheets:** Contiene las hojas de estilo en cascada, las mismas proveen de estilos visuales a las vistas del módulo.
  - **javascripts:** Contiene las funciones *javascripts*.
- ✓ **config:** Contiene el subdirectorio *locales* y el archivo *routes.rb*.

- **locales:** Se almacenan los archivos dedicados a los idiomas de la aplicación permitiendo la fácil migración a cualquiera de estos. Actualmente el módulo brinda soporte para inglés y español.
- **routes:** Se especifican las rutas de la aplicación.
- ✓ **db:** Contiene el subdirectorio *migrate*.
  - **migrate:** Contiene archivos SQL para la gestión de tablas en la base de datos. Los *migrate* permiten trabajar independiente del servidor central, y una vez terminado, es posible migrar la base de datos central.
- ✓ **lib:** Contiene clases específicas utilizadas en el módulo de RRHH.



**Figura 10** Estructura del módulo de RRHH.

### 3.1.2. Descripción de las clases

*Tabla 3 Descripción de las clases controladoras del módulo de RRHH.*

Clases controladoras (app/controllers)	
<b>BaseRrhApplicationController</b>	Clase padre de todas las clases controladoras. Contiene todos los métodos genéricos heredados por otras clases.
<b>GesproRhProjectIndicatorValuesController</b>	Manipula la información concerniente a las evaluaciones de los indicadores de los RRHH.

*Tabla 4 Descripción de la clases modelos del módulo de RRHH.*

Clases modelos (app/models)	
<b>GesproRhProjectIndicatorValue</b>	Permite la conexión directa con la tabla gespro_rh_project_indicator_values, que representa los indicadores evaluados.

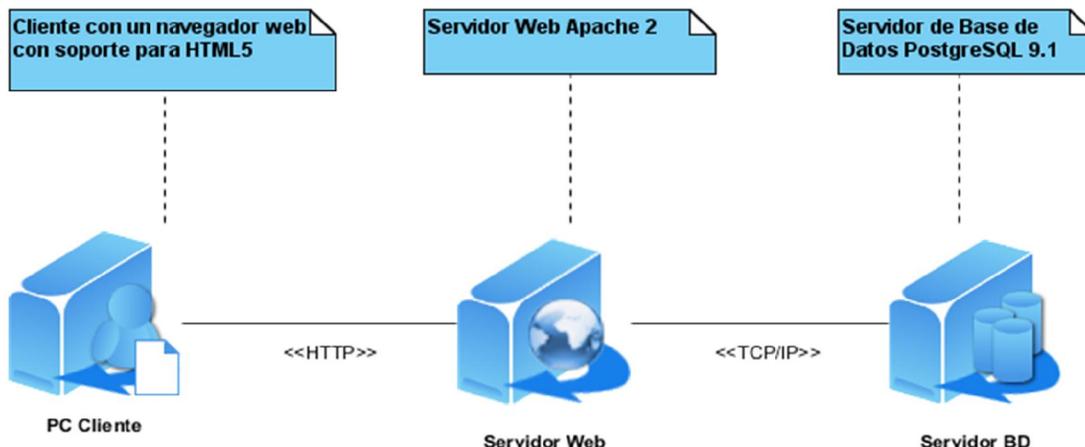
*Tabla 5 Descripción de las clases auxiliares del módulo de RRHH.*

Clases auxiliares (app/helpers)	
<b>GesproRhProjectIndicatorValueHelper</b>	Contiene métodos auxiliares para las funcionalidades asociadas a la sobrecarga de recursos humanos.

### 3.1.3. Diagrama de despliegue

En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados. Se muestran las relaciones físicas entre los componentes hardware y

software en una aplicación. Cada hardware (máquinas físicas y los procesadores) se representa como un nodo (59).



**Figura 11** Diagrama de despliegue de la Suite Xedro GESPRO.

El nodo Servidor BD está conectado con el nodo Servidor Web a través del protocolo de red TCP/IP, y éste a su vez está conectado con los nodos clientes a través de protocolos de comunicación HTTP, garantizando que en cada estación de trabajo los usuarios tengan acceso a la herramienta de manera segura. La Suite Xedro GESPRO está instalada en un servidor web Apache, que utiliza PostgreSQL como sistema gestor de base de datos.

### 3.2. Estilos de código

Un código fuente completo debe reflejar un estilo armonioso, como si un único desarrollador hubiera escrito todo el código de una sola vez. Se hace necesario establecer un estándar de codificación para garantizar la coordinación entre los desarrolladores.

Aunque la legibilidad y la estabilidad son el resultado de muchos factores, una faceta del desarrollo de software en la que todos los desarrolladores influyen especialmente es en la técnica de codificación. El mejor método para asegurarse de que un equipo de desarrollo mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código de rutinas.

Generalmente no afectan a la funcionalidad de la aplicación, pero sí contribuyen a una mejor comprensión del código fuente.

El uso de los estándares de codificación en la investigación, están presentes de la siguiente manera:

### 3.2.1. Formateo del código

Usa UTF-8 como la codificación de los archivos del código fuente.

Indentación: Usa dos espacios por cada nivel de indentación.

Tabuladores o espacios: La forma más popular de indentar en Ruby es utilizar sólo espacios. No se recomienda mezclar tabuladores y espacios.

Tamaño máximo de línea: Limita todas las líneas a un máximo de 80 caracteres.

Líneas en blanco: Separa las funciones no anidadas y las definiciones de clases con dos líneas en blanco. Las definiciones de métodos dentro de una misma clase se separan con una línea en blanco (60) (61).

```
module GesproRhIndicatorHelper

  # Esta funcion calcula el indicador irha de un rrhh hasta la fecha de corte perteneciente a un proyecto
  # return double
  def get_user_irha_by_project(user_id, project_identifier, current_date = Date.today)
    value = 0.0
    # ejecuta la llamada a un procedimiento almacenado
    value = ActiveRecord::Base.connection.select_value("SELECT * FROM _rpt_get_icp(#{user_id},#{project_identifier}',#{current_date}')")
    value.to_f.round(2)
  end

  # funcion utilizada por el plugin del cronograma para mostrar el color y estado el rrhh
  def search_indicator_and_return_color_and_tip(user_id,project_identifier)
    indicator_value = get_user_irha_by_project(user_id, project_identifier)
    color = !indicator_value.nil? ? color_by_value(indicator_value) : ''
    hash = {}
    case color
    when 'red'
      hash = {:color=>color, :tip=>1(:red_tip_label)}
    when 'green'
      hash = {:color=>color, :tip=>1(:green_tip_label)}
    when 'orange'
      hash = {:color=>color, :tip=>1(:orange_tip_label)}
    else
      hash = {:color=>color, :tip=>''}
    end
    return hash
  end
end
```

**Figura 12** Ejemplo de formateo del código.

### 3.2.2. Comentarios

Comentarios de bloque:

- ✓ Los comentarios de bloque generalmente se aplican al código que se encuentra a continuación, y se indentan al mismo nivel que dicho código. Cada línea de un comentario de bloque comienza con un # y un único espacio (a menos que haya texto indentado dentro del comentario).
- ✓ Los párrafos dentro de un comentario de bloque se separan por una línea conteniendo un único #.

Comentarios en línea:

- ✓ Utiliza comentarios en línea de forma moderada.
- ✓ Un comentario en línea es un comentario que se encuentra en la misma línea que una sentencia. Los comentarios en línea deberían separarse por al menos dos espacios de la sentencia que comentan. Deberían comenzar con un # y un espacio.

### 3.2.3. Convenciones de nombres

Estilos de nombrado:

- ✓ minusculas\_con\_guiones\_bajos.
- ✓ PalabrasEnMayusculas (CapWords o CamelCase, llamado así por el parecido de las mayúsculas con las jorobas de los camellos). Algunas veces también se llaman StudyCaps.
- ✓ capitalizacionMezclada (se diferencia de PalabrasEnMayusculas porque la primera letra está en minúsculas).

Nombres de módulos:

- ✓ Los módulos tienen nombres cortos formados en su totalidad por letras minúsculas. Se utilizan guiones bajos en el nombre del módulo si mejora la legibilidad.

Nombres de clases:

- ✓ Los nombres de clases usan la convención CapWords.

Nombres de funciones:

- ✓ Los nombres de funciones están en letras minúsculas, con palabras separadas mediante guiones bajos según sea necesario para mejorar la legibilidad.

Argumentos de funciones y métodos:

- ✓ Evita usar *self* donde no es requerido.

Nombres de métodos y variables de instancia:

- ✓ Se utilizan las reglas de los nombres de funciones: minúsculas con palabras separadas por guiones bajos cuando es necesario para mejorar la legibilidad. Se usa arroba para variables de globales (60) (61).

### 3.3. Pruebas

Las pruebas de software son elementos imprescindibles y críticos para la validación de un producto de software. Es por esto que las pruebas de software deben apoyarse en estándares que revisan los aspectos fundamentales que debe considerar todo proceso de pruebas.

El proceso de pruebas se dirige fundamentalmente a componentes del software o al sistema de software en general, con el objetivo de medir hasta cuando el software cumple las funcionalidades solicitadas por el cliente.

#### Objetivos de las pruebas

Las pruebas del software verifican y revelan la calidad de un producto. Son utilizadas para identificar posibles errores en la implementación, calidad, o usabilidad de un programa de software.

Entre los objetivos de la realización de las pruebas se presentan:

- ✓ Detectar defectos en la aplicación.
- ✓ Verificar que todos los requisitos se han implementado satisfactoriamente.
- ✓ Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el producto al usuario o cliente final.
- ✓ Diseñar los casos de prueba que permitan identificar diferentes clases de errores utilizando la menor cantidad de tiempo y esfuerzo.

#### 3.3.1. Tipos de pruebas

Las pruebas poseen una estructura en la que se define el objetivo de la prueba a realizar, la descripción detallada de la prueba y la técnica a utilizar. Entre las pruebas que se pueden encontrar se encuentran las siguientes:

- ✓ Pruebas funcionales: Estas pruebas van enfocadas a una funcionalidad completa. Las pruebas funcionales se realizan mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta la aplicación.
- ✓ Pruebas unitarias: estas pruebas se encargan de probar una clase en específico, realizando pruebas a cada uno de los métodos presentes en ella. Es una forma de

probar el correcto funcionamiento de un módulo de código de una clase. Además, permite validar que cada uno de los módulos funciona correctamente por separado.

- ✓ Pruebas de regresión: Las pruebas de regresión son una estrategia de prueba en la cual las pruebas que se han ejecutado anteriormente se vuelven a realizar en la nueva versión modificada, para asegurar la calidad después de añadir la nueva funcionalidad.
- ✓ Pruebas de validación: Proporciona una seguridad final de que el software satisface todos los requerimientos funcionales y de rendimiento. Además, valida los requerimientos establecidos comparándolos con el sistema que ha sido construido.
- ✓ Pruebas de aceptación: son un tipo de prueba funcionales, pero basadas en el cliente de la aplicación en aras de demostrar al cliente que la funcionalidad está terminada y funciona correctamente (62).

### 3.3.2. Pruebas de Caja Negra

Las pruebas de caja negra se refieren a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software (51).

Conociendo las funciones específicas para la que fue construido el producto, se pueden diseñar tales pruebas que demuestren que dichas funciones están bien realizadas. Estas pruebas son llevadas a cabo sobre las interfaces del módulo, es decir, de las funciones actuando sobre ellas como una caja negra, proporcionando unas entradas y estudiando las salidas para ver si concuerdan con las esperadas.

#### Casos de prueba

Los casos de prueba son un conjunto de condiciones mediante las cuales se comprueba si los requisitos funcionan correctamente. La descripción del caso de prueba realizado contienen el nombre del caso de uso, la descripción del caso de uso, así como la respuesta del sistema y las variables asociadas a dicho caso de uso.

**Caso de prueba #1:** Alinear la vista de presentación del módulo de RRHH mediante los principales procesos que propone la Gestión de RRHH según PMBOK.

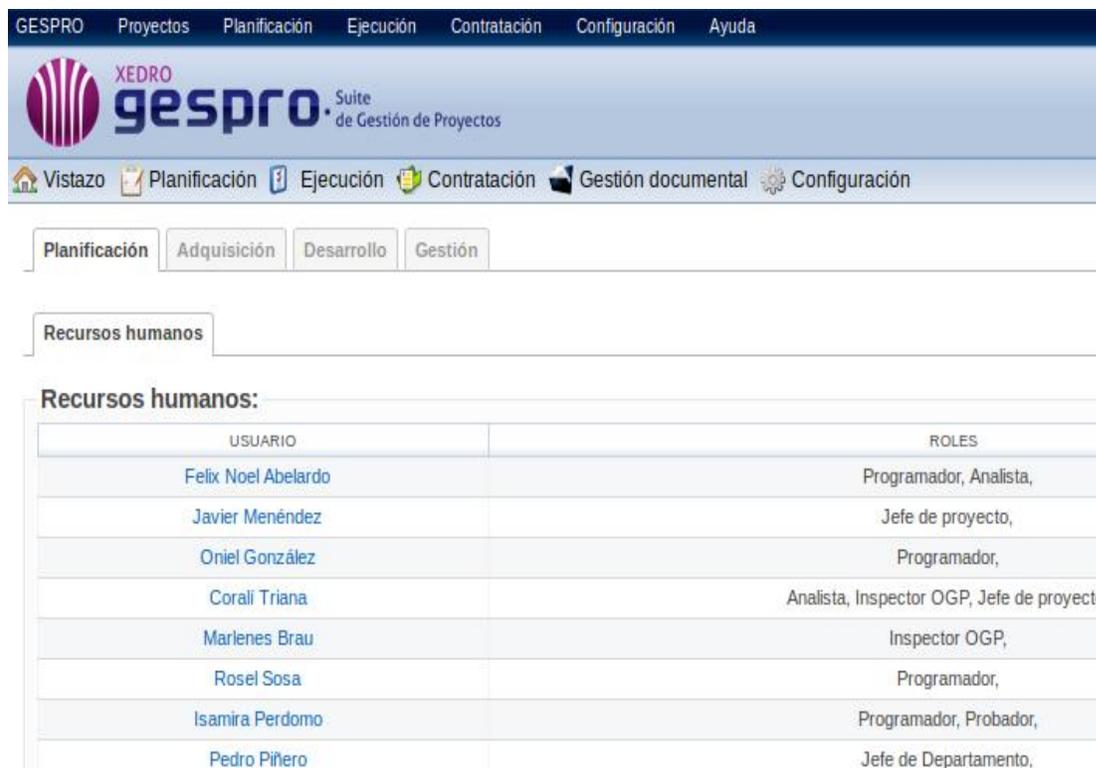
**Descripción general:** El caso de uso comienza al seleccionar la opción "Recursos Humanos" del menú desplegable de "Ejecución" en la vista de un proyecto.

**Condiciones de ejecución:** Que el usuario esté autenticado y que pertenezca a un proyecto o tengas permisos para acceder al módulo.

**Escenario:** Alinear la vista de presentación del módulo de RRHH a los principales procesos de la Gestión de RRHH.

**Tabla 6** *Mostrar Recursos Humanos.*

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1	Ver presentación del módulo de RRHH	El sistema muestra la vista de recursos humanos alineada a los principales procesos que propone la Gestión de RRHH según PMBOK.	El usuario selecciona desde la interfaz de un proyecto "Recursos Humanos" del menú desplegable de "Ejecución".



**Figura 13** *Alinear la vista de presentación del módulo de RRHH a los principales procesos de la Gestión de RRHH.*

**Caso de prueba #2:** Visualizar los recursos humanos conjuntamente con su índice de sobrecarga.

**Descripción general:** El caso de uso permite visualizar todos los recursos humanos perteneciente a un proyecto.

**Condiciones de ejecución:** Que el usuario esté autenticado y que pertenezca a un proyecto o tengas permisos para acceder al módulo.

**Escenario:** Mostar Recursos Humanos con su índice de sobrecarga.

*Tabla 7 Visualizar los recursos humanos conjuntamente con su índice de sobrecarga.*

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 2	Ver listado de recursos humanos perteneciente a un proyecto.	El sistema muestra una interfaz con los recursos humanos con su índice de sobrecarga correspondiente a un proyecto.	<ol style="list-style-type: none"> <li>1. El usuario selecciona desde la interfaz de un proyecto “Recursos Humanos” del menú desplegable de “Ejecución”.</li> <li>2. El usuario accede a la interfaz de Evaluación de Desempeño mediante opción del proceso de Gestión.</li> </ol>

#	USUARIO	FONDO T	ESTIMADO T	TAREAS	MODIFICADAS	IRHA	IRHT	IRHE	IRHF
2	Corali Triana	32	0.0	0	0	0.0	0.0	0.0	0.0
3	Felix Noel Abelardo	40	0.0	0	0	0.0	1.0	0.97	1.0
4	Isamira Perdomo	40	0.0	0	0	0.01	0.8	0.89	1.0
5	Javier Menéndez	40	0.0	0	0	0.21	0.72	0.96	0.86
6	Marlenes Brau	0	0.0	0	0	0.0	0.0	0.0	0.0
7	Oniel González	8	0.0	0	0	0.8	0.11	0.6	0.39
8	Pedro Piñero	6	0.0	0	0	0.0	0.0	0.0	0.0
9	Rosel Sosa	28	0.0	0	0	0.0	1.0	1.0	1.0

Figura 14 Visualizar los recursos humanos conjuntamente con su índice de sobrecarga (IRHA).

**Caso de prueba #3:** Colorear el índice de sobrecarga de los RRHH.

**Descripción general:** Muestra el índice de sobrecarga coloreado (Verde, Naranja o Rojo) de los RRHH.

**Condiciones de ejecución:** Que el usuario esté autenticado y que pertenezca a un proyecto o tengas permisos para acceder al módulo.

**Escenario:** Colorear el índice de sobrecarga de los RRHH.

Tabla 8 Colorear el índice de sobrecarga de los RRHH.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 3	Ver listado de recursos humanos con su índice de sobrecarga coloreado	El sistema muestra una interfaz con los recursos humanos con su índice de sobrecarga coloreado correspondiente a un proyecto.	1. El usuario selecciona desde la interfaz de un proyecto “Recursos Humanos” del menú desplegable de “Ejecución”.

	perteneciente a un proyecto.		2. El usuario accede a la interfaz de Evaluación de Desempeño mediante opción del proceso de Gestión.
--	------------------------------	--	---

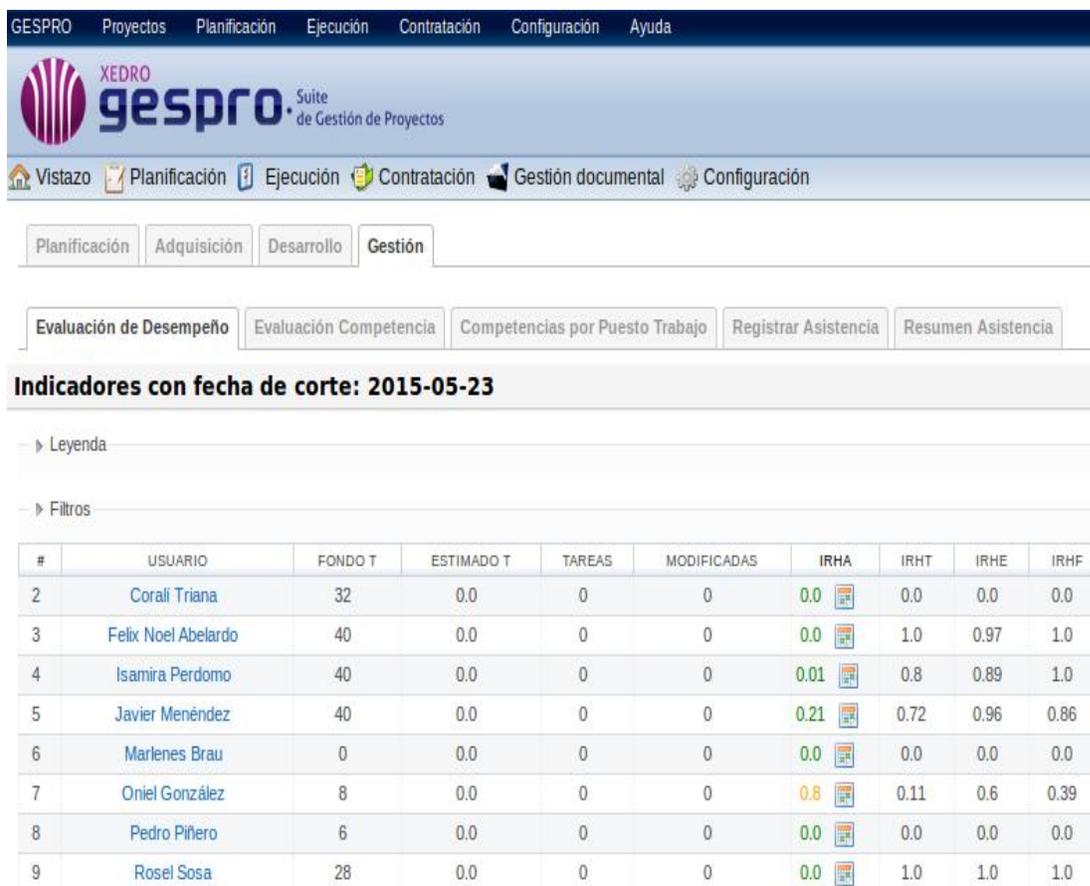


Figura 15 Colorear el índice de sobrecarga de los RRHH.

**Caso de prueba #4:** Ordenar ascendentemente o descendentemente la lista de RRHH según el índice de sobrecarga.

**Descripción general:** Muestra los RRHH ordenados por el índice de sobrecarga.

**Condiciones de ejecución:** Que el usuario esté autenticado y que pertenezca a un proyecto o tengas permisos para acceder al módulo.

**Escenario:** Ordenar ascendentemente o descendentemente la lista de RRHH según el índice de sobrecarga.

**Tabla 9** Ordenar ascendentemente o descendentemente la lista de RRHH según el índice de sobrecarga.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 4	Ver listado de recursos humanos ordenados por el índice de sobrecarga perteneciente a un proyecto.	El sistema muestra una interfaz con los recursos humanos ordenados por el índice de sobrecarga correspondiente a un proyecto.	<ol style="list-style-type: none"> <li>1. El usuario selecciona desde la interfaz de un proyecto “Recursos Humanos” del menú desplegable de “Ejecución”.</li> <li>2. El usuario accede a la interfaz de Evaluación de Desempeño mediante opción del proceso de Gestión.</li> <li>3. El usuario ordena los RRHH por el índice de sobrecarga haciendo click en la opción IRHA del listado de recursos.</li> </ol>

#	USUARIO	FONDO T	ESTIMADO T	TAREAS	MODIFICADAS	IRHA	IRHT	IRHE	IRHF
9	Rosel Sosa	28	0.0	0	0	0.0	1.0	1.0	1.0
3	Felix Noel Abelardo	40	0.0	0	0	0.0	1.0	0.97	1.0
6	Marlenes Brau	0	0.0	0	0	0.0	0.0	0.0	0.0
8	Pedro Piñero	6	0.0	0	0	0.0	0.0	0.0	0.0
2	Corali Triana	32	0.0	0	0	0.0	0.0	0.0	0.0
4	Isamira Perdomo	40	0.0	0	0	0.01	0.8	0.89	1.0
5	Javier Menéndez	40	0.0	0	0	0.21	0.72	0.96	0.86
7	Oniel González	8	0.0	0	0	0.8	0.11	0.6	0.39

**Figura 16** Ordenar ascendentemente o descendentemente la lista de RRHH según el índice de sobrecarga.

**Caso de prueba #5:** Visualizar calendario de tareas concurrentes por recurso humano.

**Descripción general:** Muestra el listado de tareas concurrentes de un recurso humano.

**Condiciones de ejecución:** Que el usuario esté autenticado y que pertenezca a un proyecto o tengas permisos para acceder al módulo.

**Escenario:** Visualizar calendario de tareas concurrentes.

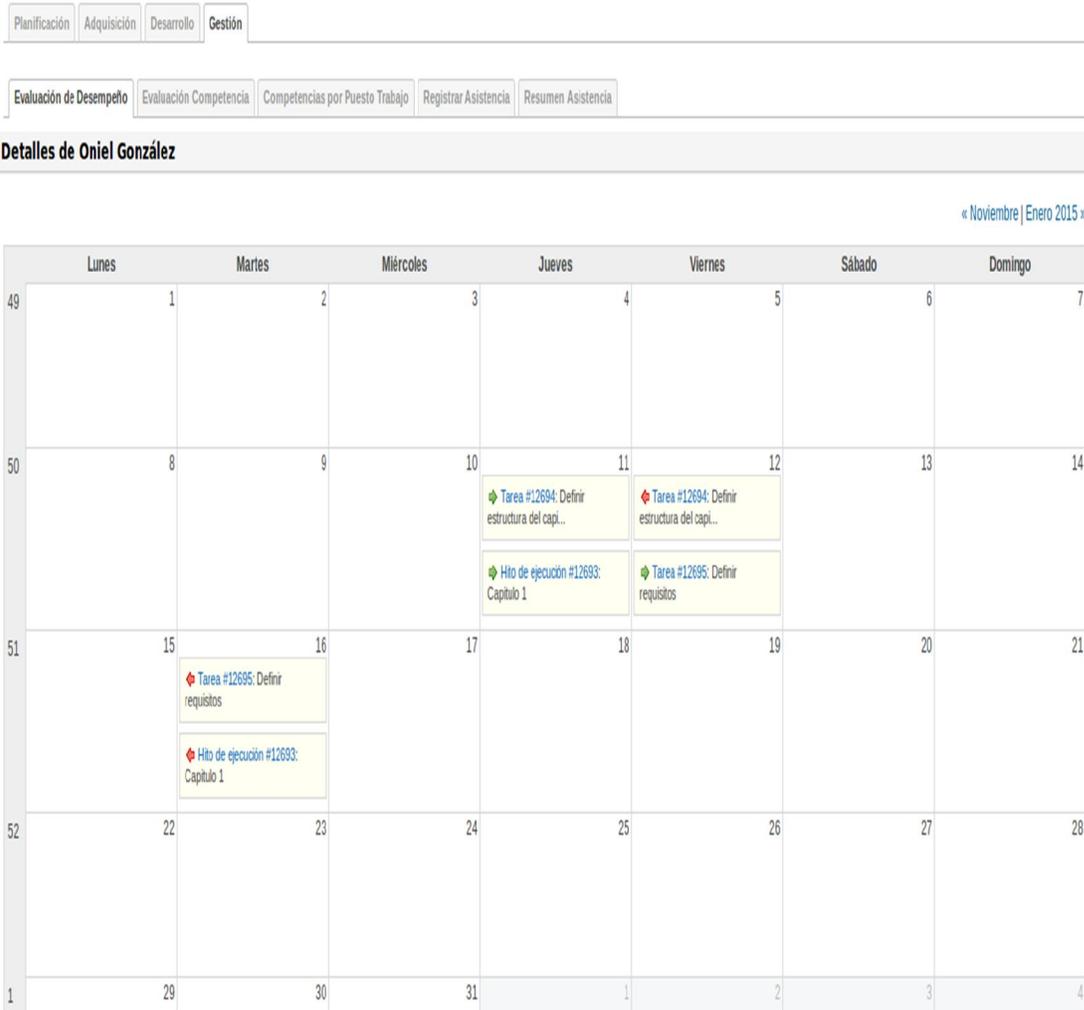
**Tabla 10** Visualizar calendario de tareas concurrentes por recurso humano.

Escenario	Descripción	ID	Respuesta del sistema	Flujo central
EC 5	Ver listado de tareas concurrentes perteneciente a un recurso humano del proyecto.	V	El sistema muestra una interfaz con los detalles de las tareas concurrentes de un recurso humano correspondiente a un proyecto.	1. El usuario selecciona desde la interfaz de un proyecto “Recursos Humanos” del menú desplegable de “Ejecución”.

				<p>2. El usuario accede a la interfaz de Evaluación de Desempeño mediante la opción del proceso de Gestión.</p> <p>3. El usuario accede a la opción “Ver detalles” haciendo click en el icono representativo del calendario en el listado de recursos.</p>
--	--	--	--	--

**Tabla 11** Descripción de los parámetros.

No	Nombre Campo	Clasificación	Valor nulo	Descripción
1	ID	Valor numérico	No	Valor del id de un recurso humano utilizado para mostrar el calendario de tareas.



**Figura 17** Visualizar calendario de tareas concurrentes.

**Caso de prueba #6:** Visualizar calendario de tareas concurrentes filtradas por mes y año, además por recurso humano.

**Descripción general:** Muestra el listado de tareas concurrentes filtradas por mes y año de las tareas asignadas a un recurso humano.

**Condiciones de ejecución:** Que el usuario esté autenticado y que pertenezca a un proyecto o tengas permisos para acceder al módulo.

**Escenario:** Visualizar calendario de tareas concurrentes filtradas por mes y año.

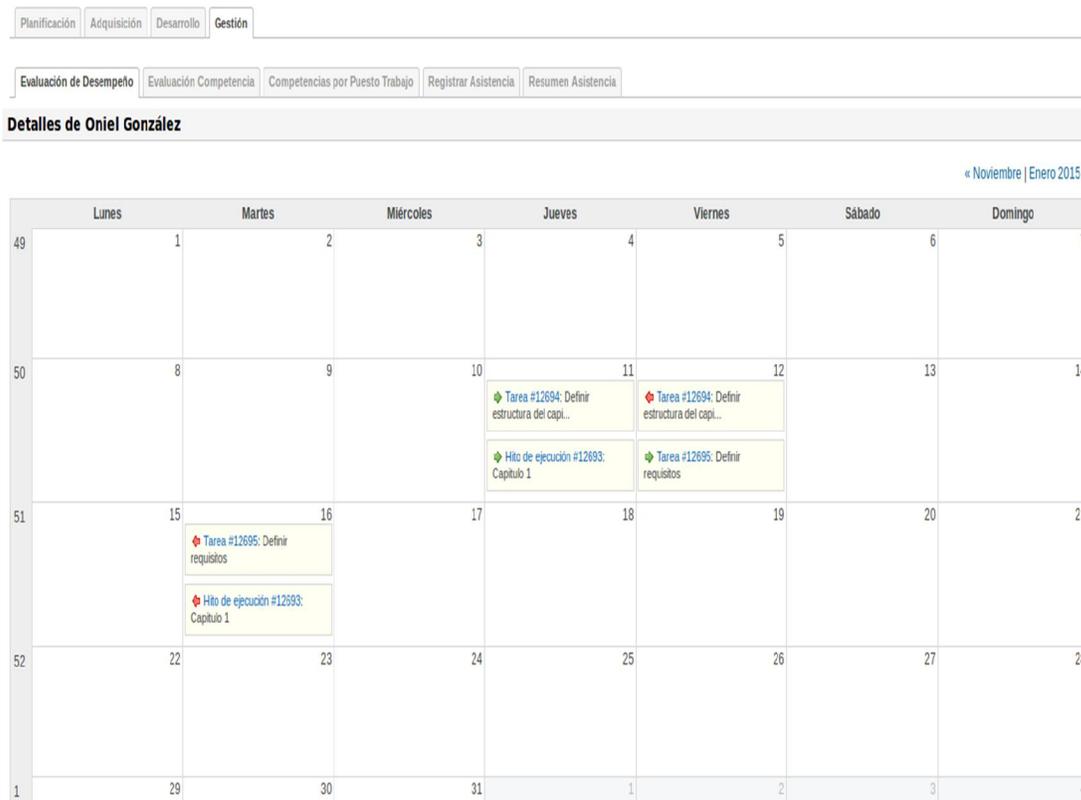
**Tabla 12** Visualizar calendario de tareas concurrentes filtradas por mes y año, además por recurso humano.

Escenario	Descripción	ID	Mes	Año	Respuesta del sistema	Flujo central
EC 6	Ver listado de tareas concurrentes filtradas por mes y año perteneciente a un recurso humano del proyecto.	V	V	V	El sistema muestra una interfaz con los detalles de las tareas concurrentes de un recurso humano filtradas por un mes y un año determinado correspondiente a un proyecto.	<ol style="list-style-type: none"> <li>1. El usuario selecciona desde la interfaz de un proyecto “Recursos Humanos” del menú desplegable de “Ejecución”.</li> <li>2. El usuario accede a la interfaz de Evaluación de Desempeño mediante la opción del proceso de Gestión.</li> <li>3. El usuario accede a la opción “Ver detalles” haciendo click en el icono representativo del calendario en el listado de recursos.</li> </ol>

						4. El usuario puede filtrar las tareas por mes y año haciendo click en el mes ubicado en la parte superior derecha del calendario.
--	--	--	--	--	--	--

**Tabla 13** Descripción de los parámetros.

No	Nombre Campo	Clasificación	Valor nulo	Descripción
1	ID	Valor numérico	No	Valor del id de un recurso humano utilizado para mostrar el calendario de tareas.
2	Mes	Valor numérico	No	Mes por el cual se filtran las tareas.
3	Año	Valor numérico	No	Año por el cual se filtran las tareas.



**Figura 18** Calendario de tareas correspondiente a un recurso humano filtradas por mes y año.

**Caso de prueba #7:** Rediseño de la vista del cronograma de tareas resaltando el uso del índice de sobrecarga.

**Descripción general:** Muestra en el *select* de “Asignado a” un listado de recursos humanos coloreados según su índice de sobrecarga, en la vista de crear una nueva tarea.

**Condiciones de ejecución:** Que el usuario esté autenticado y que pertenezca a un proyecto o tengas permisos para acceder al módulo.

**Escenario:** Rediseño de la vista del cronograma de tareas resaltando el uso del índice de sobrecarga.

**Tabla 14** Rediseño de la vista del cronograma de tareas resaltando el uso del índice de sobrecarga.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 7	Ver listado de recursos humanos resaltando su	El sistema muestra una interfaz con los detalles para crear una nueva tarea o modificar una	1. El usuario selecciona desde la interfaz de un proyecto “Cronograma”

	índice de sobrecarga.	existente resaltando el “asignado a” con un listado de recursos humanos pertenecientes a un proyecto.	del menú desplegable de “Planificación”. 2. El usuario accede a la interfaz de Crear una nueva tarea mediante opción Agrega ubicado en la parte superior izquierda del cronograma.
--	-----------------------	---	---

**Caso de prueba #8:** Mostrar mensaje de disponibilidad: “Disponible”, “No disponible” o “Disponible con riesgo” correspondiente a cada recurso humano de un proyecto.

**Descripción general:** Muestra un mensaje de disponibilidad: “Disponible”, “No disponible” o “Disponible con riesgo” al seleccionar el recurso humano al cual se le desea asignar la tarea.

**Condiciones de ejecución:** Que el usuario esté autenticado y que pertenezca a un proyecto o tengas permisos para acceder al módulo.

**Escenario:** Mostrar mensaje de disponibilidad: “Disponible”, “No disponible” o “Disponible con riesgo” correspondiente a cada recurso humano de un proyecto.

**Tabla 15** *Mostrar mensaje de disponibilidad: “Disponible”, “No disponible” o “Disponible con riesgo” correspondiente a cada recurso humano de un proyecto.*

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 8	Ver mensaje de disponibilidad de cada recurso humano del proyecto.	El sistema muestra un <i>tooltip</i> con el mensaje de disponibilidad: “Disponible”, “No Disponible” o “Disponible con riesgo” correspondiente a un recurso humano perteneciente a un proyecto.	1. El usuario selecciona desde la interfaz de un proyecto “Cronograma” del menú desplegable de “Planificación”. 2. El usuario accede a la interfaz de Crear una nueva tarea mediante opción Agrega ubicado en la parte superior

			<p>izquierda del cronograma.</p> <p>3. El usuario selecciona de la lista desplegable “Asignado a” el recurso humano al que se le va a asignar la tarea.</p>
--	--	--	---

General | Avanzado | Grupo | Precedencia

Tipo \*: Tarea

Descripción:

Estado \*: Nueva

Prioridad \*: Baja

Asignado a \*: Oniel González

Iteración prevista: Coralí Triana

Tarea padre: Felix Noel Abelardo

Isamira Perdomo

Tiempo Estimado(h) \*: Javier Menéndez

Tiempo Dedicado: Oniel González

Disponible con riesgo

Pedro Piñero

Pedro Piñero Pena

Rosel Sosa

(Gestionar Tiempo Dedicado)

**Figura 19** Mostrar mensaje de disponibilidad: “Disponible”, “No disponible” o “Disponible con riesgo” correspondiente a cada recurso humano de un proyecto.

### 3.3.3. Resultados

A continuación se muestran los resultados obtenidos a partir de la aplicación de pruebas de caja negra basado en casos de prueba, las cuales fueron realizadas en 3 iteraciones, donde arrojaron varias no conformidades que fueron erradicadas en el transcurso de las mismas. En la primera iteración se detectaron 2 no conformidades, en la segunda 1 no conformidad, esta última fue completamente resuelta para en una tercera iteración tener un resultado favorable de 0 no conformidades.



### 3.4. Conclusiones parciales

En el presente capítulo se detallaron los elementos correspondientes a la fase de implementación, donde se obtuvo como resultado una nueva versión del módulo de RRHH de la Suite Xedro GESPRO. Además se representó el diagrama de despliegue y se logró la aplicación de técnica de validación permitiendo asegurar que los requerimientos del módulo se describan de manera correcta.

## CONCLUSIONES GENERALES

Una vez culminada la investigación es posible afirmar que se cumplieron todos los objetivos trazados para la misma:

- ✓ El estudio de los procesos de la disciplina Gestión de Recursos Humanos permitió comprender y aplicar las teorías existentes sobre la sobrecarga de recursos humanos al Módulo de Recursos Humanos en la Suite Xedro GESPRO.
- ✓ La implementación de las funcionalidades incorporadas al Módulo de Recursos Humanos contribuyeron a mejorar la planificación y análisis de recursos humanos en la Suite Xedro GESPRO.
- ✓ La aplicación de las pruebas al Módulo de Recursos Humanos, contribuyó a mejorar el funcionamiento y alcanzar la satisfacción del usuario con el uso de la Suite Xedro GESPRO.

## RECOMENDACIONES

- ✓ Añadir la funcionalidad IRHA al Módulo Peticiones de la Suite Xedro GESPRO, para contribuir a una mejor planificación de los RRHH en los proyectos.
- ✓ Incorporar las nuevas funcionales del Módulo de Recursos Humanos en la ayuda de la Suite Xedro GESPRO, para facilitar la utilización de la herramienta.

## REFERENCIAS BIBLIOGRÁFICAS

1. **Torres López, Surayne.** 2011. *Modelo para la Gestión de los Recursos Humanos en Centros de Desarrollo de Sistemas de Información.* La Habana, Cuba : Universidad de las Ciencias Informáticas.
2. **Partido Comunista de Cuba (PCC).** 2011. *Lineamientos de la Política Económica y Social del Partido y.* La Habana : PCC.
3. **Real Academia Española.** 2001. *Diccionario de la lengua española.* [En línea] [Citado el: 8 de enero de 2015.] <http://lema.rae.es/drae/?val=recurso>.
4. **Fernández, J. M.** 1999. *Recursos humanos: Fundamentos del comportamiento humano en la empresa.* Madrid, España : Ediciones Encuentro.
5. **Chiavenato, I.** 2007. *Administración de Recursos Humanos. El Capital Humano de las Organizaciones.* México DF : McGraw-Hill. págs. 357-392. ISBN 958-41-0037-8.
6. **Definición.de.** *Concepto de gestión — Definición.de.* [En línea] <http://definicion.de/gestion/>.
7. **PMI.** 2009. *Guía de los Fundamentos para la Dirección de Proyectos (PMBOK).* Pennsylvania, EEUU : PMI Publications.
8. **Rodríguez Lagual, Yndra Lucia.** "GESTIÓN DE RECURSOS HUMANOS". [En línea] <http://www.eoi.es/blogs/madeon/2013/03/10/gestion-de-recursos-humanos/>.
9. **Project Management Institute.** *What is Project Management? | Project Management Institute.* [En línea] <http://www.pmi.org/About-Us/About-Us-What-is-Project-Management.aspx>.
10. **ADR Formación.** *Recursos sobreasignados e infraasignados en MS Project - Ofimática - MS Project - Artículos ADR Formación.* [En línea] [http://www.adrformacion.com/articulos/ms\\_project/recursos\\_sobreasignados\\_e\\_infraasignados\\_en\\_ms\\_project/articulo761.html](http://www.adrformacion.com/articulos/ms_project/recursos_sobreasignados_e_infraasignados_en_ms_project/articulo761.html).
11. **aulaClic.** *Curso gratis de Microsoft Project 2007. aulaClic. 6 - Sobreasignación y redistribución.* [En línea] [http://www.aula clic.es/project-2007/t\\_6\\_1.htm](http://www.aula clic.es/project-2007/t_6_1.htm).
12. **Delgado Victore, R.** 2003. *La Dirección Integrada de Proyectos haciendo uso de las Nuevas Tecnologías de la Informática y las Comunicaciones.* La Habana : CETA ISPJAE.

13. **OpenProject Foundation.** *Open Project.* [En línea] <https://community.openproject.org/projects/openproject/wiki/FAQ>.
14. **Microsoft.com.** *Microsoft Project.* [En línea] <http://www.microsoft.com/project>.
15. **DotProject.net.** 2011. *Comunidad dotProject.* [En línea] <http://www.dotproject.net>.
16. **Collabtive.com.** *Collabtive Project Management and Open Source Groupware.* [En línea] <http://www.collabtive.com>.
17. **Collabtive.** *Collabtive.* [En línea] <http://collabtive.o-dyn.de>.
18. **Atlassian.com.** 2011. *Atlassian Jira.* [En línea] <http://www.atlassian.com/software/jira/overview>.
19. **Teamwork.com.** *Teamwork Project Management.* [En línea] <https://www.teamwork.com/>.
20. **Lang, J P.** *Redmine.org.* [En línea] <http://www.redmine.org>.
21. **SEI.** 2010. *CMMI for Dev v1.3.* Pittsburgh : Carnegie Mellon University.
22. **Ecured.** *Xedro Gespro.* [En línea] <http://www.ecured.cu/index.php/Gespro>.
23. **GESPRO.** 2012. *Repositorio de Código Fuente GESPRO.* [En línea] <http://git.dt.prod.uci.cu>.
24. **Piñero Pérez, P. Y. y Colectivo de Autores.** 2013. *GESPRO. Paquete para la gestión de proyectos.* Cuba : Revista Nueva Empresa. págs. 45-53. Vol. 9. ISSN 1682-2455.
25. **Rodríguez Figueredo, H.** 2009. *El movimiento del software libre en Cuba.* s.l. : Revista Cubana de Ciencias Informáticas. págs. 5-13.
26. **Ecured.** *Metodologías de desarrollo de software.* [En línea] [http://www.ecured.cu/index.php/Metodologías\\_de\\_desarrollo\\_de\\_software](http://www.ecured.cu/index.php/Metodologías_de_desarrollo_de_software).
27. **Cockburn, Alistair.** 2000. *Agile Software Development.* s.l. : Highsmith Series.
28. **Carnós, José H. (et. al.).** 2003. *Metodologías Ágiles en el Desarrollo de Software.* Alicante - España : Grupo ISSI.
29. **Cohen, Mike.** Una Introducción a Scrum. [En línea] Noviembre de 2013. <http://www.dc.fi.udc.es/~cabalar/md/scrum.pdf>.

30. **Rodríguez González, Pilar.** 2009. *Estudio de la aplicación de metodologías ágiles para la evolución de productos software.* La Habana,Cuba : s.n.
31. **Schwaber, K.** 2010. *Advanced Development Methods. SCRUM Development Process.* [En línea] <http://www.jeffsutherland.com/scrumpapers.pdf>.
32. **R. C., Martin, M., Beedle y Schwaber, K.** 2001. *Agile Software Development with SCRUM.* Prentice Hall : s.n.
33. **Ubuntumx.** 2012. *Que es Ubuntu.* [En línea] <http://www.ubuntumx.org/>.
34. **Ubuntu.** 2012. *Filosofía Ubuntu.* [En línea] [http://doc.ubuntu-es.org/Filosof%C3%ADa\\_Ubuntu](http://doc.ubuntu-es.org/Filosof%C3%ADa_Ubuntu).
35. **Dpto. de Sistemas de Informáticos y Computación.** *Introducción a Herramientas CASE y System Architect.* UNIVERSIDAD POLITÉCNICA DE VALENCIA.
36. **UML CASE.** UML CASE tool for software development. [En línea] <http://www.visual-paradigm.com/product/vpuml/>.
37. **Rumbaugh, James, J, I y Booch, Gragy.** 2000. *El Lenguaje Unificado de Modelado.* Madrid : s.n. ISBN 84-7829-037-0.
38. **Franganillo, Jorge.** HTML5: el nuevo estándar básico del web. [En línea] [Citado el: 29 de Enero de 2014.] <http://thinkipi.net/html5-nuevo-estandar-basico-del-web>.
39. **w3.** What is CSS? *World Wide Web Consortium.* [En línea] [Citado el: 29 de Enero de 2014.] <http://www.w3.org/standards/webdesign/htmlcss#whatcss>.
40. **W3C.** HTML & CSS - W3C. [En línea] <http://www.w3.org/standards/webdesign/htmlcss#whatcss>.
41. **Ruby.** 2011. *Ruby a Programmer's Best Friend.* [En línea] <https://www.ruby-lang.org/es/about/>.
42. **Ruby on Rails.** *Ruby on Rails 3.2 Release Notes — Ruby on Rails Guides.* [En línea] [http://guides.rubyonrails.org/3\\_2\\_release\\_notes.html](http://guides.rubyonrails.org/3_2_release_notes.html).
43. **PostgreSQL Global Development Group.** 2011. *PostgreSQL.org.* [En línea] <http://www.postgresql.org>.

44. **Ecured.** *Sistema de Gestión de Base de Datos.* [En línea] [http://www.ecured.cu/index.php/Sistema\\_Gestor\\_de\\_Base\\_de\\_Datos](http://www.ecured.cu/index.php/Sistema_Gestor_de_Base_de_Datos).
45. **Comunidad de PostgreSQL.** *Acerca de PostgreSQL.* [En línea] [Citado el: 16 de Febrero de 2015.] [http://postgresql.uci.cu/?page\\_id=30](http://postgresql.uci.cu/?page_id=30).
46. **PgAdmin3.** 2011. *PgAdmin3 PostgreSQL Tools.* [En línea] <http://www.pgadmin.org/>.
47. **Apache.** *Apache Http Server.* [En línea] [Citado el: 19 de Enero de 2014.] <http://httpd.apache.org/>.
48. **NetBeans.** NetBeans IDE 6.9 Release Information. [En línea] <https://netbeans.org/community/releases/69/index.html>.
49. **Lugo García, José Alejandro.** 2012. *Modelo para el Control de la Ejecución de Proyectos basado en Indicadores y Lógica Borrosa.* La Habana, Cuba : Universidad de las Ciencias Informáticas.
50. **Hernández Alfonso, Arlan.** 2012 *Análisis y diseño del módulo de gestión de los recursos humanos para el sistema GESPRO.* La Habana,Cuba : Universidad de las Ciencias Informáticas.
51. **Pressman, Roger S.** 2005. *Ingeniería de Software: Un enfoque práctico.* Sexta Edición. s.l. : McGraw-Hill. págs. 165-180,183, 191-244. Vol. II. ISBN:9701054733.
52. **Larman, Craig.** 2000. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* 1ra Edición. s.l. : Prentice Hall, Mexico. ISBN 970-1 7-0261-1.
53. **Hall, C.L.P.** 2005. *Applying UML and Patterns. An introduction to Object-Oriented Analysis and Design and Iterative Development.* 3rd Edition.
54. **Mesa, Jose Luis.** 2012. *Patrón de Acceso a datos (DAO).*
55. **Ramos, Gastón.** 2010. *Active Record, sabor Ruby.*
56. **Heinemeier Hansson, David.** Active Record Basics — Ruby on Rails Guides. [En línea] [http://guides.rubyonrails.org/active\\_record\\_basics.html](http://guides.rubyonrails.org/active_record_basics.html).
57. **Yiiframework.** 2010. Working with databases. [En línea] <http://www.yiiframework.com/doc/guide/1.1/es/database.ar>.

58. **Mar Cornelio, Omar.** 2012. *Application for the power control technology using network monitoring tool Nmap. Ingeniería y gestión de software.* Universidad de las Ciencias Informáticas. La Habana : s.n. pág. 10. ISSN: 1994-1536.
59. **Valdivia Cerda, Anibal Sayid.** 2003. *Modelo de diseño en UML e implantación del Sistema de Gestión para las dependencias de la universidad de Colimas.* Ingeniería Mecánica y Eléctrica, Universidad de Colima. Colima : s.n. Tesis de Maestría.
60. **GitHub.** *Ruby on Rails 3 & 4 style guide.* [En línea] <https://github.com/bbatsov/rails-style-guide>.
61. **Github.** *Manual de estilo para Ruby.* [En línea] <https://github.com/Rojo/Manual-de-estilo-para-Ruby>.
62. **EcuRed.** *Pruebas de Calidad de Software.* [En línea] [http://www.ecured.cu/index.php/Pruebas\\_de\\_Calidad\\_de\\_Software](http://www.ecured.cu/index.php/Pruebas_de_Calidad_de_Software).
63. **Zadeh, L. A.** 1994 *Fuzzy Logic, Neural Networks and Soft Computing, in Fuzzy Logic, Neural Networks and Soft Computing. Communications of the ACM.* págs. 77-84. Vol. 37.
64. **Wikipedia.** *Framework.* [En línea] <http://es.wikipedia.org/w/index.php?title=Framework&oldid=80876102>.