Universidad de las Ciencias Informáticas

Centro de Informatización Universitaria

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Sistema de Registro y Control de Personal UCI

Autoras: Yunet Labrada Alpizar

Mayelin Illana Alméciga

Tutores:

Ing. Jorge L. Tamarit Cutiño

Ing. Alberto Tamayo Ramos

Consultantes:

Ing. Aranay Otero Cutiño

Tec. Ariadna Falcón Novelo

Ciudad de La Habana, junio 2010

"Año 52 de la Revolución"

RESUMEN

El trabajo de diploma tiene como objetivo mejorar los procesos de registro de personal en la Universidad de las Ciencias Informáticas (UCI). Actualmente en este centro existe un Sistema de Registro Personal, pero no responde a las necesidades actuales del negocio, por lo que se desarrolla un sistema capaz de gestionar el registro y control del personal tercero, familiar y visitante de la UCI.

El desarrollo del sistema está guiado por procesos, utilizando como lenguajes de modelado Lenguaje de Modelado Unificado (UML) y Notación para el Modelado de Procesos de Negocio (BPMN), las metodologías utilizadas son Scrum y XP. Como Entorno Integrado de Desarrollo (IDE) se utiliza NetBeans. Se usa PHP 5 como lenguaje de programación y CodeIgniter 1.7.2 como framework de desarrollo, que implementa el patrón de arquitectura Modelo Vista Controlador (MVC) y como gestor de bases de datos se emplea PostgreSQL 8.4.

El Sistema de Registro y Control de Personal UCI contribuye a la mejora de los servicios asociados a los procesos de registro de personal, proporcionando información actualizada y accesible en tiempo real, además brinda diferentes reportes asociados al negocio que se maneja.

Palabras Claves: Registro y Control de Personal, personal tercero, personal familiar, personal visitante.

ÍNDICE

INTRODUCCIÓN **1**

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA **5**

1.1 Objetivos Estratégicos de la Organización. **5**

1.2 Los Sistemas de Registro y Control de Personal **5**

1.3 Sistemas de gestión de información de recursos humanos existentes en el mundo. **5**

1.3.1 A nivel internacional *5*

1.3.2 A nivel nacional *6*

1.3.3 A nivel institucional *7*

1.4 Tecnologías, metodologías y herramientas utilizadas **8**

1.4.1 Arquitectura del sistema **8**

1.4.2 Tecnologías utilizadas en el proceso de desarrollo **8**

1.4.2.1 Tecnologías del lado del cliente *8*

1.4.2.2 Tecnologías del lado del servidor *10*

1.4.3 Metodología de desarrollo de software **11**

1.4.3.1 Lenguaje de modelado *12*

1.4.4 Herramientas utilizadas **13**

1.4.5 Servicios Web **14**

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA **15**

2.1 Flujo actual de los procesos involucrados en el campo de acción **15**

2.1.1 Descripción del flujo actual de los procesos *15*

2.2 Análisis crítico de la ejecución de los procesos **16**

2.3 Objeto de automatización **17**

2.4 Lista de Reserva del Producto (LRP) **18**

2.5 Personas involucradas en el sistema **23**

2.6 Historias de Usuario **24**

2.6.1 Descripción de las historias de usuario *24*

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN **37**

3.1 Descripción de la arquitectura **37**

3.2 Estrategia de integración **38**

3.3 Modelo de datos **38**

3.4 Tareas generales de la implementación (TI) **40**

3.5 Estándar de codificación **43**

3.6 Servicios brindados **44**

3.7 Diagrama de despliegue 44
CONCLUSIONES 46
RECOMENDACIONES 47
REFERENCIAS BIBLIOGRÁFICAS 48
BIBLIOGRAFÍA 50
ANEXOS 53
GLOSARIO DE TÉRMINOS 66

INTRODUCCIÓN

Con el transcurso de los años, en el mundo han evolucionado los avances científicos y tecnológicos. Cuba a pesar de ser un país subdesarrollado, se ha incorporado al desarrollo y uso de las tecnologías, pues aprecia la importancia que tiene su aplicación para lograr progresos considerables en el orden económico, político y social del país.

Una de las tendencias más abundantes en el uso de las tecnologías, es proporcionar soluciones informáticas a aquellos procesos que forman parte del quehacer cotidiano en cualquier centro laboral. Estas soluciones están encaminadas al desarrollo de sistemas, dirigidos a automatizar procesos de gran importancia para la sociedad y cuyo objetivo es alcanzar mayor eficiencia en la producción de los servicios, mejor calidad de vida y satisfacer las necesidades en las diferentes esferas.

Con el propósito de extender el desarrollo informático hacia todos los lugares del país, se está llevando desde hace algunos años un intenso proceso de informatización, en el que la UCI, proyecto de la revolución cubana, surgida al calor de la batalla de ideas, se destaca como centro de vanguardia en el desarrollo de software.

En la universidad habitan diferentes tipos de personas, estas personas de acuerdo con la actividad que ejercen en el centro están categorizadas en cinco grandes grupos.

1. Estudiantes: son todas aquellas personas que realizan sus estudios universitarios en la UCI.
2. Trabajadores UCI: personal que es plantilla del centro. Tales como: profesores, auxiliares de servicio, instructoras, secretarias docentes, técnicos, directivos, especialistas, asistentes y choferes.
3. Terceros: personal que pertenece a empresas y entidades externas como es el personal de salud, mantenimiento, gastronomía, seguridad y servicios técnicos, además, está el personal de otras áreas terceras como: Softel, Albet, MININT y FAR.
4. Familiares: son todas aquellas personas que residen en la universidad por cuestiones familiares como: hijos, nietos, padres, esposas (os), hermanos y otros familiares.
5. Visitantes: son personas que visitan el centro por diferentes razones por un corto período de tiempo, existen tres tipos de visitas: familiares, laborales y VIP (persona muy importante).

El registro de estas personas en los inicios de la universidad se realizaba de forma manual en documentos Excel, propiciando la pérdida de valiosa información, además de arduas horas de trabajo, de aquí la importancia de automatizar todo el proceso de registro de personas en el centro.

Primeramente se solucionó el registro de estudiantes y trabajadores UCI en el proceso de prematrícula-matrícula y en el proceso de contratación respectivamente. De manera tal que el registro y control de terceros, familiares y visitantes es hoy una de las principales necesidades de la UCI.

Para estos tres grupos de persona inicialmente se realizó un intento de implementación centrado en la gestión de información, desarrollado en software propietario, resolviendo el problema en ese momento. A medida que fueron surgiendo cambios en las políticas del negocio, se incrementaron sus necesidades, lo que trajo consigo que esta solución ya no era óptima para cumplir con las expectativas requeridas.

Posteriormente se realizó un nuevo sistema que está vigente actualmente en el centro, el cual mantiene la prioridad de registro del personal tercero en específico, dejando fuera y descartando en su totalidad la variante del registro de visitantes y familiares. Este desarrollo dio un paso de avance respecto al anterior, pues está implementado en software libre.

El constante cambio y dinamismo en el que se encuentra envuelta la universidad ha propiciado que este sistema no responda a las necesidades actuales del negocio que se maneja. Entre las principales deficiencias que presenta se encuentran las siguientes:

- No brinda reportes.
- No cumple con las pautas de arquitectura de información, arquitectura de software y diseño que plantea el Centro de Informatización Universitaria para lograr una identidad UCI.
- Falta de maniobrabilidad e interacción con otras aplicaciones necesarias para el buen funcionamiento del sistema.
- No presenta una correcta accesibilidad por roles.
- Permite la duplicación de personas en el sistema.
- La implementación del código fuente no cuenta con una estructura legible que le permita a los desarrolladores un fácil mantenimiento.
- No presenta una base de datos normalizada.

Luego de analizar la problemática planteada y las dificultades existentes en la forma que es gestionada la información del personal tercero, familiar y visitante en la universidad se identifica como **problema científico**: ¿Cómo contribuir al perfeccionamiento del flujo de procesos referentes a la gestión de información del personal tercero, familiar y visitante?

El problema planteado se enmarca en el **objeto de estudio**: los procesos de gestión de personal. El objeto de estudio delimita el **campo de acción:** los procesos que se realizan en el registro del personal tercero, familiar y visitante en la Universidad de las Ciencias Informáticas.

Por lo antes planteado se definió como **objetivo de la investigación**: Desarrollar una solución que posibilite la gestión del personal tercero, familiar y visitantes y que permita la obtención óptima de diferentes reportes.

Para lograr el objetivo de la investigación planteado se trazaron las siguientes **tareas de la investigación:**

- Establecer en detalles el actual flujo de procesos para el registro del personal tercero, familiar y visitante en la UCI.
- Entrevistar a los posibles clientes para identificarse con las nuevas necesidades de funcionamiento en el proceso actual.
- Recopilar diferentes plantillas de registro por categorías de persona en las diferentes áreas responsables del personal tercero, familiar y visitante.
- Valorar sobre la existencia y funcionamiento de posibles aplicaciones de este tipo ya creadas.
- Realizar una propuesta de arquitectura para el desarrollo de dicha solución.
- Realizar una propuesta de modelo de datos como soporte básico sobre el cual se implementará el sistema.
- Identificar los posibles reportes que pudieran formar parte del futuro enriquecimiento de la solución esperada.
- Identificar los requisitos de software funcionales y no funcionales necesarios para dar cumplimiento a la implementación del sistema.
- Diseñar la solución propuesta.
- Implementar la solución propuesta.

- Desarrollar una capa de servicio que permita brindar información necesaria al sistema de Identificación y a la Intranet 2.0.
- Realizar la documentación que explique todo el proceso de desarrollo de dicho sistema una vez terminada su implementación.

El presente trabajo está estructurado en 3 capítulos que resumen la siguiente información:

Capítulo 1. Fundamentación Teórica. Se define el objeto de estudio, se realiza un análisis de los sistemas de gestión de información de recursos humanos existentes en el mundo, asociado al campo de acción y se describen las herramientas, tecnologías y metodologías que se utilizan en el desarrollo del sistema propuesto.

Capítulo 2. Características del Sistema. Se realiza una concepción inicial del negocio con el objetivo de comprender el contexto del sistema. Se define la solución propuesta, los procesos que son objeto de automatización. Se muestra la Lista de Reserva del Producto (LRP), el plan de iteración y se describen las historias de usuario.

Capítulo 3. Diseño e Implementación. Se describe la arquitectura del sistema, los servicios web que son brindados y los que son usados desde otros sistemas. Es mostrada la distribución física de la aplicación, además el modelo de datos que contiene la descripción de cada una de sus entidades. Se describen las tareas de ingeniería y se define el estándar de codificación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el siguiente capítulo se realiza un estudio del arte sobre los sistemas existentes en el mundo, vinculados al campo de acción. Se definen las tecnologías, herramientas y metodologías que se usan para el desarrollo del sistema, en función de las tendencias actuales.

1.1 Objetivos Estratégicos de la Organización

Mantener un control administrativo y una correcta organización sobre todas aquellas personas que ingresan a la universidad, desarrollan actividades y brindan servicios a la comunidad universitaria, pero que no forman parte del volumen de personas propias de la entidad, dígase estudiantes o trabajadores UCI.

1.2 Los Sistemas de Registro y Control de Personal

Los Sistemas de Registro y Control de Personal tienen como tarea fundamental almacenar y procesar información, centrando su atención en el control y planificación. Estos sistemas constituyen una poderosa herramienta de trabajo que le permite a cualquier centro mantener el control de todos los trabajadores que laboran en el mismo, optimizando así sus recursos. En la actualidad existen diversos sistemas de gestión de recursos humanos, que posibilitan el control y que son necesarios en la organización interna de cualquier entidad o empresa laboral. [1]

1.3 Sistemas de gestión de información de recursos humanos existentes en el mundo

1.3.1 A nivel internacional

Empleados MSD Multiusuario: Es un avanzado gestor de información, que permite manejar datos personales, familiares, historial, documentos y titulaciones entre otros servicios. Este sistema está estructurado por cuatro módulos: Módulo Empleados, Módulo Información de Empleados, Módulo Resumen y Módulo Estadísticas. Es una aplicación fácil de utilizar y permite configurar muchas de sus características para adaptarse a los gustos del usuario. Brinda un alto grado de privacidad y aporta herramientas de suma utilidad tales como: copias de seguridad, procesador de textos, calculadora y otras. [2]

Wikimetal RRHH: Es un sistema destinado a la gestión del personal de una empresa, de simple y fácil manejo que ha sido creada con el fin de ayudar a las empresas en la difícil tarea del control de sus empleados. Entre otras funciones, brinda la posibilidad de gestionar las vacaciones de los empleados de forma mensual, bimensual o trimestral. Puede efectuar un control por departamentos e incluso visualizar inmediatamente los retrasos, ausencias o sanciones de los empleados. [3]

@GestRRHH versión 2.0: Es un sistema diseñado para la gestión de Recursos Humanos compuesto de los siguientes módulos: agenda, plantilla, actividades y nóminas. Entre las principales funcionalidades que presenta están las siguientes: permite la realización de un gran número de informes, cada módulo posee un color individual para mejorar la información y permite la copia de seguridad automática. [4]

Estos sistemas no se ajustan a las políticas y necesidades del sistema a desarrollar, porque ofrecen una visión de como es gestionada la información de los recursos humanos de forma general.

1.3.2 A nivel nacional

Registro Personal de la Salud (RPS): Es un sistema que está basado en software libre, en las nuevas tecnologías y en el uso de servicios web que garantizan una explotación integrada y compatible. Utiliza como soporte de esta aplicación la infraestructura de la Red Telemática de Salud (INFOMED). Cumple con la arquitectura definida por el Grupo de Arquitectura MINSAP–MIC y hace uso de la Plataforma de Servicios (PlaSer). [5]

Este sistema tiene las siguientes características:

- Almacena datos en una base de datos única, sólida y confiable.
- Realiza la actualización del Registro Personal de la Salud (profesionales y técnicos) de manera rápida y segura.
- Permite gestionar al personal de salud pendiente de ubicar en una unidad de salud, abarcando los movimientos de traslado, jubilación y baja, así como los movimientos de los recién graduados.
- Busca profesionales por sus características, que puede ser: nombre, tipo de profesional y nivel técnico.

Selsoft: Es un producto diseñado para la evaluación integral de los requisitos exigidos por el perfil psicológico en el proceso de Selección de Personal y Evaluación de los puestos de trabajo. La elaboración de los perfiles psicológicos se realiza con el usuario. La herramienta ha sido desarrollada utilizando el Lenguaje Visual Basic Versión 3.0 y Access 2.0 como gestor de bases de datos. [6]

Fastos: Es otro sistema en el ámbito nacional que gestiona la información de los recursos humanos, conformado por módulos que ejecutan los procesos y operaciones de: registro de empleados, control de la plantilla, control de asistencia, informes y modelos. [7]

Estos sistemas presentan como deficiencia, que la gestión de información que manejan difiere con lo planteado en la problemática y en algunos casos están implementados con tecnología obsoleta.

1.3.3 A nivel institucional

ASSETS (Sistema de Registro Integral): Sistema que registra todo el personal que es plantilla en la UCI, este sistema es del tipo de Planificación de Recursos Empresariales (ERP), de factura foránea. Se introdujo en Cuba desde el año 1997. Controla plenamente los recursos laborales, dígase: empleados, estructura organizativa de la entidad y plantilla. Este sistema solo gestiona el personal que es plantilla del centro por lo que no sirve de ayuda a la hora de encontrarle una solución a la problemática planteada.

Sistema de Registro de Personal (SRP): Este sistema está enmarcado en el registro del personal tercero de la UCI. Fue desarrollado usando el lenguaje de programación PHP, PostgreSQL como gestor de base de datos y el servidor web utilizado es Apache. SRP tiene varias funcionalidades como: gestionar personas, atributos por categoría, categorías de persona y roles.

Esta solución a pesar de mantenerse en función y de cierto modo realizar el registro del personal tercero específicamente que es uno de los tipos de persona con necesidad de registro en la universidad, no es factible porque no responde al flujo de procesos que resuelve verdaderamente el problema.

1.4 Tecnologías, metodologías y herramientas utilizadas

1.4.1 Arquitectura del sistema

MVC: Es un patrón de arquitectura de software que se aplica frecuentemente en el desarrollo de aplicaciones web. Este patrón separa los datos, la interfaz de usuario y la lógica de control en tres componentes distintos, donde:

- La vista transforma el modelo en una página web que permite al usuario interactuar con ella e interpretar el código que provee de datos dinámicos a la página.
- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Cliente-Servidor: La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. [8]

1.4.2 Tecnologías utilizadas en el proceso de desarrollo

1.4.2.1 Tecnologías del lado del cliente

JavaScript: Es utilizado principalmente en la construcción de páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. JavaScript al igual que Java, es un lenguaje orientado a objetos que dispone de herencia, esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript como recurso para tener una mayor maniobrabilidad del Document Objet Model (DOM). [9]

HTML: Siglas de Hyper Text Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, para complementar el texto con objetos de imágenes. HTML se escribe

en forma de "etiquetas", rodeadas por corchetes angulares (<,>), también puede hasta un cierto punto ayudar a lograr la apariencia de un documento. [10]

CSS: Siglas de Cascading Style Sheets (Hojas de Estilo en Cascada), es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe como se va a mostrar un documento en pantalla. [11]

Ajax*:* Siglas de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o Aplicaciones de Internet Enriquecidas (RIA). Estas aplicaciones se ejecutan en el cliente. Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado, en el que normalmente se efectúan las funciones de llamada, mientras que el acceso a los datos se realiza mediante XMLHttpRequest (Extensible Markup Language / Hypertext Transfer Protocol), objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML. [12]

Ajax es una combinación de cuatro tecnologías ya existentes: [12]

- XHTML o HTML y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- DOM accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar incluyendo HTML preformateado, texto plano, JavaScript Object Notation (JSON) y hasta Extensible Binary Meta Language (EBML).

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y DOM.

1.4.2.2 Tecnologías del lado del servidor

PHP: Representa Hypertext Preprocessor (Preprocesador de Hipertexto), es un lenguaje script que se ejecuta del lado del servidor en la Arquitectura Cliente – Servidor. PHP es muy utilizado por los desarrolladores de páginas web por ser uno de los lenguajes de programación más completo, siendo capaz de manejar un entorno que integre gestores de bases de datos como: MySQL, Microsoft SQL Server, PostgreSQL, SQLite y Oracle, permitiendo la creación de aplicaciones web muy robustas. Tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como los soportados por UNIX, Linux, Mac OS y Windows, además puede interactuar con los servidores web más populares. Es muy legible y fácil de aprender, no soporta directamente punteros, de forma que no existen los problemas de depuración provocados por estos y es libre por lo que es de fácil acceso para todos.[13]

CodeIgniter: Es un framework que proporciona una buena base para implementar fácilmente aplicaciones web y además es muy rápido ya que permite actualizar dinámicamente el contenido. CodeIgniter podría no ser la mejor opción para desarrollar aplicaciones muy grandes y complejas; es más adecuado para proyectos pequeños o medianos.

Algunas características importantes de CodeIgniter: [14]

- Compatible tanto para PHP 4 como para PHP 5.
- Magnifica documentación y enorme comunidad de desarrolladores.
- Facilidad: No requiere más que una media hora de estudio para empezar a hacer cosas interesantes.
- Usa el patrón de diseño Modelo-Vista-Controlador como paradigma de arquitectura de desarrollo, la cual separa en 3 capas distintas: la representación de datos, el interfaz de usuario y el controlador de eventos respectivamente.

Servidor HTTP Apache: Es uno de los servidores web más populares del mercado, y el más utilizado actualmente, es de código abierto y gratuito, disponible para Windows y GNU/Linux. Apache presenta mensajes de error altamente configurables, bases de datos de autenticación y un negociador de contenido. [15]

Entre sus características fundamentales se puede destacar:

- Es flexible, rápido y eficiente.
- Multiplataforma.
- Se desarrolla de forma abierta.
- Modular: Puede ser adaptado a diferentes entornos, a las necesidades de los diferentes módulos de apoyo que proporciona y con la API (Aplication Programming Interface) de programación de módulos para el desarrollo de módulos específicos.
- Incentiva la realimentación de los usuarios obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.

PostgreSQL: Es un Sistema Gestor de Bases de Datos (SGBD) relacionales orientadas a objetos. Es el gestor de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguaje de programación (incluyendo C, C++, Java, PERL y Python). Es capaz de manejar una gran cantidad de datos, permitiendo numerosos accesos simultáneos de los usuarios.

1.4.3 Metodología de desarrollo de software

Para obtener mejores resultados en la organización y planificación del trabajo, así como un producto informático con calidad, que satisfaga tanto a clientes y a desarrolladores, es preciso usar una metodología de desarrollo de software.

Se decide optar por una metodología ágil para el desarrollo del software debido al poco tiempo de desarrollo y el reducido grupo de trabajo. Los principales objetivos que persigue esta metodología son:

- La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- Entregar frecuentemente software que funcione desde un par de semanas a un par de meses con el menor intervalo de tiempo posible entre entregas.
- El software que funciona es la medida principal de progreso.
- La atención continua a la calidad técnica y al buen diseño mejora la agilidad.

SCRUM_XP: Es una metodología ágil compuesta por SCRUM y XP (Extreme Programming) que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles mejorando la

actividad productiva, fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo y ayudando al líder del proyecto a ejercer mejor control sobre el mismo.

SCRUM se encarga de gestionar un equipo para que trabaje de forma eficiente y medir siempre los procesos. XP es una metodología encaminada al desarrollo, consiste en una programación rápida o extrema cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Esta metodología consta de cuatro fases principales:

1. Definición-Planificación: es donde se establece la visión, se fijan las expectativas y asegura el financiamiento del proyecto.
2. Desarrollo: se realiza la implementación del sistema y se obtiene un producto listo para ser entregado.
3. Entrega: se transfiere el producto al cliente.
4. Mantenimiento: donde se realiza el soporte para el cliente.

SCRUM_XP está especialmente indicada para proyectos de pequeños equipos de trabajo, con requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta una entrega rápida de resultados y una alta flexibilidad. Ayuda a que el equipo trabaje unido, en la misma dirección y con un objetivo claro. Permite seguir de forma clara el avance de las tareas a realizar, ayudando a los líderes a ver día a día como funciona el trabajo.

1.4.3.1 Lenguaje de modelado

BPMN: Es una notación de modelado de procesos de negocio que permite crear una red de objetos gráficos, que son actividades (trabajo) y controles de flujo que definen su orden de rendimiento. Esta notación permite ser comprendida fácilmente por todas las personas involucradas en el negocio desde el analista hasta el desarrollador quien es el encargado de implementar la tecnología que llevarán a cabo dichos procesos. [16]

UML: Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales de los

procesos de negocio y funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar. [17]

1.4.4 Herramientas utilizadas

NetBeans: Es un Entorno Integrado de Desarrollo (IDE) para escribir, compilar, depurar y ejecutar programas. NetBeans permite que las aplicaciones se desarrollen a partir de un conjunto de módulos o componentes de software. Es fácil de instalar y de uso instantáneo, se ejecuta en varias plataformas incluyendo Windows, Linux y Mac OS y Solaris. También es un producto libre y gratuito sin restricciones de uso permitiendo crear aplicaciones con diferentes lenguajes de programación como PHP, Java y otros.

Visual Paradigm – UML: Es una de las herramientas CASE del mercado más completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. Visual Paradigm-UML también proporciona características tales como: generación del código, ingeniería inversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y viceversa. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. [18]

BizAgi Process Modeler: Es una herramienta que permite realizar diagramas y documentar procesos de negocio (flujogramas) de una forma fácil, intuitiva y ágil, basándose en el estándar de notación BPMN, en otras palabras, es un modelador de procesos que permite representar de forma esquemática todas las actividades y decisiones que se toman en el negocio. [19]

Axure: Es una aplicación ideal para crear prototipos y especificaciones muy precisas para páginas web. Cuenta con todo lo que se puede necesitar para crear los prototipos de forma más eficiente. Permite componer la página web visualmente, añadiendo, quitando y modificando los elementos con suma facilidad.

EMS Manager for PostrgreSQL: Es una poderosa herramienta gráfica para la administración del servidor de bases de datos PostgreSQL. Funciona con cualquier versión de PostgreSQL. Ofrece una gran cantidad de poderosas herramientas a usuarios experimentados para satisfacer todas sus necesidades. PostgreSQL Manager cuenta con una novedosa interfaz gráfica de usuario con un sistema asistente bastante descriptivo. [20]

1.4.5 Servicios Web

Un servicio web es una colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos.

WSDL: Es un Lenguaje de Descripción de Servicios Web basado en XML que permite la descripción de los servicios desplegados por una aplicación.

SOAP: Siglas de Simple Object Access Protocol, que es un protocolo estándar que define la manera en que dos objetos en diferentes procesos, pueden comunicarse por medio de datos XML.

Conclusiones

En el capítulo se han demostrado las facilidades que brindan los sistemas de recursos humanos empleados en determinadas empresas, manifestándose así la necesidad que presenta la universidad de contar con un sistema de este tipo, que gestione el personal tercero, familiar y visitante. Es importante destacar que la selección de las tecnologías, metodologías y herramientas se hizo teniendo en cuenta la propuesta del cliente, en este caso, el Centro de Informatización Universitaria.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

El siguiente capítulo plantea una descripción del sistema que se pretende desarrollar. Se identifican y definen los procesos del negocio, la Lista de Reserva del Producto (LRP), la descripción de las historias de usuario y se define el plan de iteración.

2.1 Flujo actual de los procesos involucrados en el campo de acción

Realizar un análisis detallado del flujo de procesos, permite conocer el funcionamiento del negocio a través de cual se realizará una solución que responda a las necesidades del cliente.

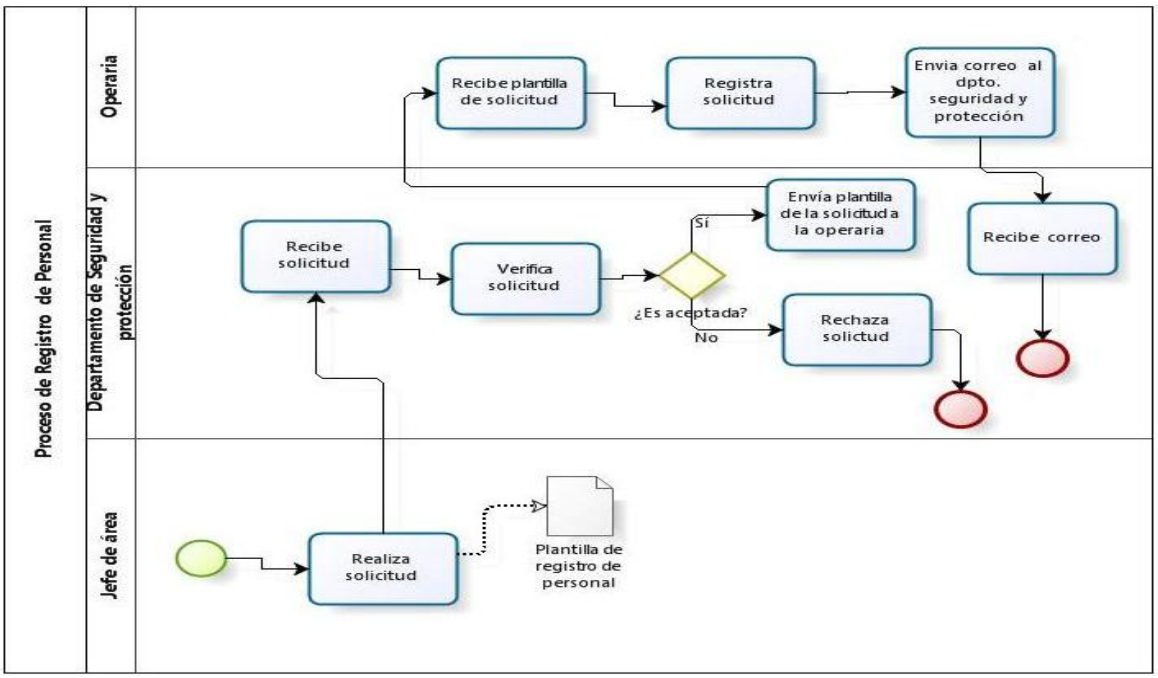


Figura 1: Proceso de Registro de Personal

2.1.1 Descripción del flujo actual de los procesos

Ficha de Proceso	
Proceso:	Registro de Personal
Entradas:	Datos básicos pertenecientes al personal tercero o eventual (carné de identidad), categoría de persona, área a la que pertenece y el cargo que ocupa, recogidas en la plantilla de registro.

Salidas:	Id-expediente generado para cada persona.
Involucrados:	Departamento de seguridad y protección y la operaria.
Descripción: El registro de personal es un proceso que permite registrar oficialmente un personal tercero o eventual.	
1. Realizar la solicitud	El jefe de área realiza la solicitud manualmente llenando la plantilla de registro de personal y envía esta plantilla mediante el correo electrónico al departamento de seguridad y protección.
2. Recibe solicitud.	Después de realizada una solicitud el departamento de seguridad y protección recibe la solicitud.
3. Verificar solicitud	La solicitud es revisada por el departamento de seguridad y protección. 3.1 Si la solicitud es aceptada se le envía un correo con los datos de la persona solicitada a la operaria para que realice el registro en el sistema. 3.2 Si la solicitud es rechazada se termina el proceso.
Regla del negocio: Cuando la solicitud es realizada para registrar un eventual se verifica que la fecha de fin de contrato esté correcta.	

Tabla 1: Descripción del flujo actual de los procesos

2.2 Análisis crítico de la ejecución de los procesos

Teniendo en cuenta que el flujo actual del proceso referente al registro de personal ha posibilitado satisfacer las necesidades básicas en la universidad, se pueden mencionar algunos aspectos deficientes que se han detectado y que han generado parte de la problemática que se pretende solucionar con la confección del presente trabajo. No debe existir un intermediario entre el sistema y el departamento de seguridad y protección, no se implementa un sistema de seguridad poniendo en peligro la confidenciabilidad de los datos en el proceso de registro de personal y no se le hace llegar al solicitante el resultado de la verificación de su solicitud.

2.3 Objeto de automatización

Con el objetivo de perfeccionar y eliminar los problemas que han sido identificados en el análisis crítico del proceso actual, se pretende automatizar el siguiente proceso.

- **Proceso de registro de personal**

Se inicia cuando el solicitante (el director del área o la asistente de residencia) selecciona el tipo de solicitud que desea realizar, el sistema muestra la plantilla correspondiente con una serie de campos a llenar, cuando se introducen los datos en esta plantilla y se pulsa el botón aceptar, la solicitud adquiere el estado pendiente a revisión. El aprobador es el encargado de verificar la solicitud, para realizar este procedimiento, primeramente el sistema muestra un listado donde se encuentran todas las solicitudes pendientes, luego de seleccionar alguna de estas, se visualizan todos los datos correspondientes a la misma para su revisión, cuando se realiza dicha acción el aprobador puede tomar una de las tres decisiones que se describen a continuación:

- Si la información solicitada está correcta, la solicitud es aceptada y obtiene el estado de aprobada y se le envía una notificación de aceptación al solicitante activándose la persona en el sistema.
- En caso de que alguno de los datos solicitados no estén correctos, entonces la solicitud es aplazada y se le envía una notificación al solicitante (incluye un campo de observación donde se indican los datos que deben ser corregidos para que la solicitud sea aceptada), posteriormente este debe seleccionar del listado la solicitud correspondiente, es decir, la que tiene el estado aplazada, seguidamente se muestran los datos de la misma luego de culminarse la modificación de la solicitud, comienza nuevamente el proceso de verificación debido a que al concluir esta actividad la solicitud obtiene el estado de pendiente a revisión.
- Si la persona solicitada no cumple con los requisitos necesarios entonces la solicitud es denegada y se le envía una notificación al solicitante argumentándole la razón por la cual su solicitud fue denegada.

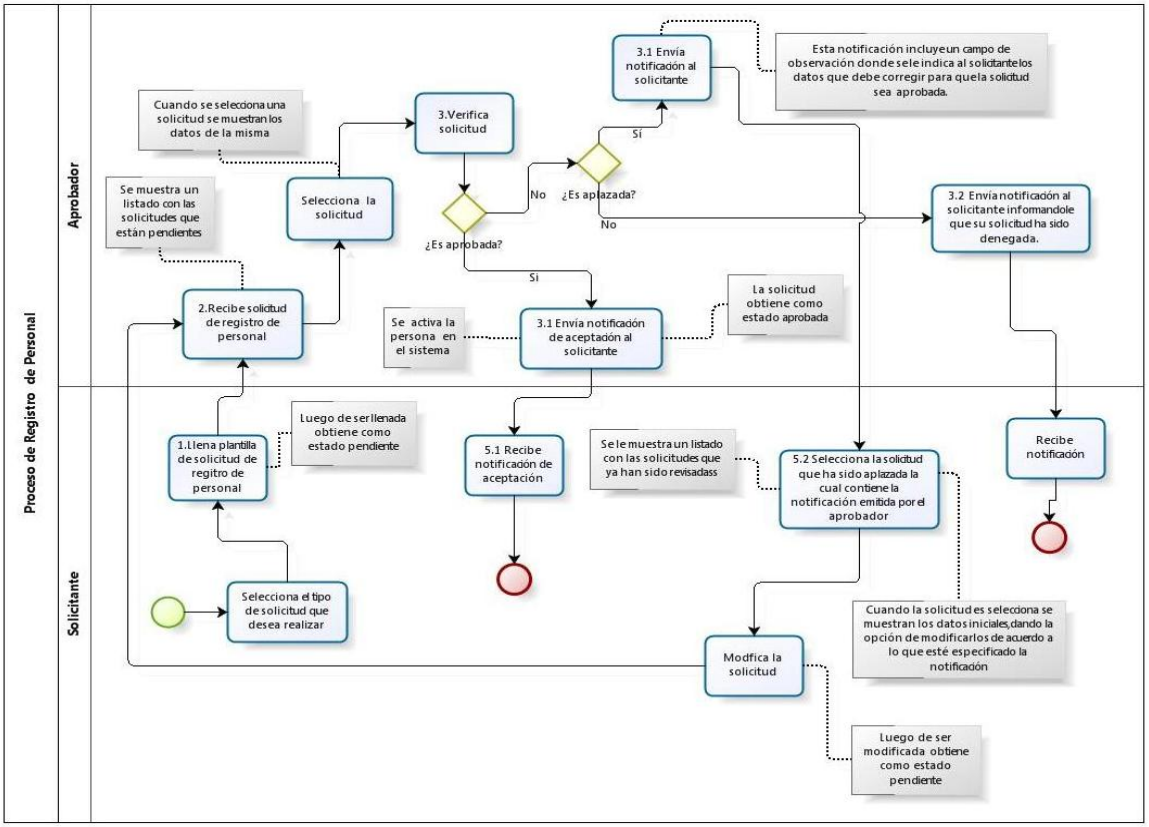


Figura 2: Proceso a automatizar

2.4 Lista de Reserva del Producto (LRP)

Es una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requisitos sobre el producto. Esta lista puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente. El objetivo es asegurar que el producto definido al terminar la lista sea el correcto, útil y lo más competitivo posible, para esto la lista debe acompañar los cambios en el entorno y el producto.

Requerimientos Funcionales	
1.	Autenticar usuario
Solicitud	
Gestionar solicitud	

2.	Buscar persona solicitante
3.	Realizar solicitud
4.	Buscar solicitud
5.	Listar solicitudes
6.	Verificar solicitud
7.	Modificar solicitud
Gestionar registro	
8.	Buscar persona
9.	Listar personas
10.	Modificar datos de persona
11.	Dar alta
12.	Dar baja
Seguridad	
Gestionar usuario	
13.	Insertar usuario
14.	Modificar usuario
15.	Activar usuario
16.	Desactivar usuario
17.	Listar usuarios
Gestionar nomenclador rol	
18.	Crear nomenclador rol
19.	Modificar nomenclador rol
20.	Activar nomenclador rol

21.	Desactivar nomenclador rol
22.	Listar nomenclador rol
Configuración	
Gestionar atributos de persona	
23.	Insertar atributo de persona
24.	Modificar atributo de persona
25.	Desactivar atributo de persona
26.	Listar atributos de persona
Gestionar nomenclador área tercera	
27.	Crear nomenclador área tercera
28.	Modificar nomenclador área tercera
29.	Activar nomenclador área tercera
30.	Desactivar nomenclador área tercera
31.	Listar nomenclador área tercera
Gestionar nomenclador categoría de persona	
32.	Crear nomenclador categoría de persona
33.	Modificar nomenclador categoría de persona
34.	Activar nomenclador categoría de persona
35.	Desactivar nomenclador categoría de persona
36.	Listar nomenclador categoría de persona
Estadísticas	
Gestionar reportes	
37.	Obtener solicitudes por fecha

38.	Obtener personal por estado de persona
39.	Obtener personal por categoría de persona
Requerimientos no funcionales	
1. Usabilidad	
1.1 Facilidad de uso por parte de los usuarios: el sistema deberá presentar una interfaz amigable con el objetivo de que el usuario pueda llegar de manera rápida y efectiva a la información buscada. 1.2 Emplear perfiles de usuario: diferenciar las interfaces y opciones para los usuarios que accedan al sistema según los diferentes roles que estos tengan dentro del sistema. 1.3 El sistema deberá presentar un menú lateral que permita el acceso rápido y cómodo a la información por parte de los usuarios.	
2. Fiabilidad	
2.1 Seguridad de la base de datos: La base de datos deberá estar fraccionada en esquemas que permitan un mejor uso de la información y la división de forma lógica de las funcionalidades del sistema, trayendo consigo la protección de la información al ocurrir un incidente sobre una parte de la base de datos. El SGBD escogido debe presentar facilidades de administración de roles y usuarios, restringiendo el acceso a los datos. 2.2 Servicios web restringidos: los servicios web brindados por el sistema deben estar restringidos a grupos de usuarios definidos y aprobados previamente. 2.3 Políticas de seguridad por usuarios y roles: el sistema debe contar con un grupo de políticas de accesibilidad a las diferentes funcionalidades del mismo en dependencia del nivel de autorización que presente un usuario determinado. 2.4 Alta protección de los datos: al estar trabajando con información sensible, se hace necesario una alta protección de los datos entre la aplicación y el tráfico de red, para tal fin se ha definido además, la seguridad en varios niveles dentro de la aplicación: nivel de interfaz, nivel de acceso a datos y nivel de base de datos.	
3. Eficiencia	

3.1 El sistema debe soportar un tiempo de respuesta menor o igual a 5 segundos. 3.2 El sistema debe soportar una conexión simultánea de más de 100 usuarios.
4. Soporte
4.1 Grupo de soporte: el sistema debe contar con un grupo de soporte al cliente del producto.
5. Restricciones de diseño
5.1 Lenguaje de programación: PHP 5.1 o superior 5.2 El marco de trabajo base de desarrollo que se utilizará es: CodeIgniter 1.7.2 5.3 Como IDE se empleará NetBeans 6.7. 1 5.4 Como servidor web se explotará Apache 2.2.2. 5.5 El SGDB deberá ser PostgreSQL 8.4.1. 5.6 El diseño de la base de datos se realizará con Visual Paradigm 3.0. 5.7 El sistema operativo a utilizar en el entorno de desarrollo deberá ser: GNU Linux 5.8 El repositorio principal, el entorno de prueba y el servidor de bases de datos estarán montados sobre Ubuntu Server 9.04 o superior
6. Requisitos para la documentación de usuarios en línea y ayuda del sistema
6.1 Manual de usuario: el sistema deberá presentar un manual de usuario, permitiendo con ello un correcto uso de sus funcionalidades y brindarle al usuario una mayor experiencia del trabajo con el mismo. 6.2 Documentación actualizada del grupo de desarrollo: se precisa que la documentación del sistema esté actualizada en todos los aspectos, fases de trabajo y ciclos de desarrollo del mismo, permitiendo con ello un respaldo tanto ingenieril como legal del desarrollo de dicho sistema.
7. Interfaz
7.1 Interfaz Web: la interfaz deberá ser sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo de su empleo.

8. Hardware
8.1 Para el desarrollo: PC Intel Pentium 4 o superior, CPU 3GHZ o superior, 512 MB RAM o superior, 160 GB HDD o superior. 8.2 Para explotación del cliente: PC Pentium 3 o superior, CPU 133 MHZ o superior, 256 RAM mínimo 512 RAM recomendada o superior. 8.3 Para explotación del servidor: CPU Dual Core 2.0 GHZ o superior, memoria RAM de 4 GB (recomendado 6 GB), 250 GB de disco duro.
9. Requisitos Legales, de Derecho de Autor y otros
9.1 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 que solicitó el producto.

Tabla 2: Lista de Reserva del Producto

2.5 Personas involucradas en el sistema

Roles del sistema	Descripción
Administrador del sistema	Es el encargado de garantizar la seguridad y la configuración del sistema.
Director del área	Es la persona que realiza las solicitudes de visitantes VIP, visitantes laborales y terceros. Además es el encargado de las operaciones administrativas del registro y de las solicitudes asociadas a seguridad y protección.
Director de residencia	Es el encargado de verificar las solicitudes de familiares y visitas familiares, ejerciendo el rol de aprobador.
Director de seguridad y protección	Es el encargado de verificar las solicitudes de

	visitantes VIP y visitantes laborales, además de las solicitudes de terceros.
Asistente de residencia	Es la persona que realiza las solicitudes de familiares y visitantes familiares. Es la encargada de las operaciones administrativas del registro y de las solicitudes asociadas a la dirección de residencia.

Tabla 3: Personas involucradas en el sistema

2.6 Historias de Usuario

Las historias de usuario (HU) son la técnica utilizada en XP para especificar los requisitos del software. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definirlas. Son escritas en lenguaje natural y proveen detalles suficientes para hacer una estimación razonable del tiempo que llevará implementarlas.

2.6.1 Descripción de las historias de usuario

2.6.1.1 Historia de usuario Autenticar usuario

Historia de usuario	
Nombre historia de usuario: Autenticar usuario.	
Número: 01	**Referencia:** 1
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 1
Prioridad: Alta	**Puntos Estimados**: 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 10 horas
Descripción: El usuario entra sus datos (usuario y contraseña) en el formulario, el sistema verifica que los datos entrados estén correctos y que se trate de un usuario con privilegios, en caso correcto se le permite el acceso, en caso contrario se le informa un mensaje de aviso indicándole la negativa del acceso.	
Observaciones: Los usuarios deben tener el mínimo privilegio absoluto necesario para realizar las tareas asignadas.	

Prototipo de interfaz: Anexo 1

Tabla 4: HU Autenticar usuario

2.6.1.2 Historia de usuario Buscar persona solicitante

Historia de usuario	
Nombre historia de usuario: Buscar persona solicitante	
Número: 02	**Referencia:** 2
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 2
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 12 horas
Descripción: La HU permite realizar la búsqueda de la persona solicitante, esta se inicia cuando se selecciona el tipo de solicitud que se desea realizar dentro del bloque realizar solicitudes. Seguidamente el sistema muestra un campo para que se introduzca el criterio de búsqueda (nombre, número de carné de identidad, usuario y número del solapín), luego se listan todas las personas que cumplen con el criterio introducido y el usuario debe de dar clic encima de la persona buscada para que se muestre una vista con los datos esperados.	
Observaciones: En caso de que no exista al menos una persona que cumpla con el criterio de búsqueda, el listado aparece vacío.	
Prototipo de interfaz: Anexo 2	

Tabla 5: HU Buscar persona solicitante

2.6.1.3 Historia de usuario Realizar solicitud

Historia de usuario	
Nombre historia de usuario: Realizar solicitud	
Número: 03	**Referencia:** 03
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 2
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 20 horas

Descripción: La HU se inicia luego de concluir con la búsqueda de la persona solicitante donde se muestra una vista que contiene los datos de la misma. El usuario debe de seleccionar el tipo de persona que se desea solicitar y el sistema sobrecarga la vista que devuelve la búsqueda de la persona solicitante agregándole, los campos que se necesitan llenar para solicitar dicha persona. Luego de introducir los datos y pulsar el botón aceptar, el sistema emite un mensaje de confirmación.
Observaciones: • En el caso de que los datos introducidos estén incorrectos se muestra un mensaje de información y se señalan en rojo los campos que el usuario debe de corregir para poder realizar la solicitud. • Cuando se realiza una solicitud esta obtiene el estado pendiente.
Prototipo de interfaz: Anexo 3

Tabla 6: HU Realizar solicitud

2.6.1.4 Historia de usuario Buscar solicitud

Historia de usuario	
Nombre historia de usuario: Buscar solicitud	
Número: 04	**Referencia:** 04, 05
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 2
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 20 horas
Descripción: La HU se inicia cuando se selecciona la opción solicitudes del bloque operaciones administrativas o cuando se selecciona la opción verificar del bloque verificar solicitudes. El sistema muestra una vista donde se listan todas las solicitudes realizadas y permite realizar la búsqueda de una o varias solicitudes en específico, mostrando un selector que permite filtrar por criterios (fecha y categoría de persona), además de un campo donde se pueden introducir otros criterios de búsqueda de la persona solicitada (nombre y carné de	

identidad). Cuando se pulsa el botón buscar el sistema muestra un listado con todas las solicitudes que cumplen con los criterios.
Observaciones: • En el caso de verificar la información que contiene el listado que se muestra inicialmente y luego de introducir los criterios de búsqueda, solamente se visualizan las solicitudes en estado pendiente y que cumplan con los criterios establecidos por el usuario. Luego de dar clic sobre la solicitud buscada, se muestra una vista con la plantilla de dicha solicitud.
Prototipo de interfaz: Anexo 4 y Anexo 5

Tabla 7: HU Buscar solicitud

2.6.1.5 Historia de usuario Verificar solicitud

Historia de usuario	
Nombre historia de usuario: Verificar solicitud	
Número: 05	**Referencia:** 06, 11
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 2
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 14 horas
Descripción: La HU se inicia al concluir la búsqueda de la solicitud. Luego de culminar con la verificación de los datos de la plantilla de solicitud, el usuario puede seleccionar una de las cuatro opciones que se proponen: • Es en el caso que no exista ningún error en la plantilla de solicitud se selecciona la opción aceptar, el sistema le envía una notificación al solicitante (incluye la observación) y la solicitud obtiene el estado de aprobada. • En caso de que exista un error en los datos que no sea crítico, se comenta el campo de observación con los datos a corregir para que la solicitud sea aceptada y se selecciona la opción aplazar. El sistema envía la observación al solicitante y la solicitud obtiene el estado de aplazada.	

• En caso de que la persona solicitada no cumpla con los requisitos necesarios, el usuario debe de comentar el campo de observación introduciendo los motivos de la decisión tomada, el sistema envía notificación al solicitante y la solicitud obtiene el estado de denegada. • En caso pulsar el botón cancelar el sistema muestra la vista inicial correspondiente a la opción verificar del bloque verificar solicitudes.
Observaciones: Cuando una solicitud es aceptada la persona obtiene el estado de alta.
Prototipo de interfaz: Anexo 6

Tabla 8: HU Verificar solicitud

2.6.1.6 Historia de usuario Modificar solicitud

Historia de usuario	
Nombre historia de usuario: Modificar solicitud	
Número: 06	**Referencia:** 07
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 2
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 16 horas
Descripción: La HU se inicia al concluir la búsqueda de la solicitud, seleccionando dicha solicitud y dando clic en el ícono que representa la opción modificar solitud, ubicado a la derecha del listado en la barra de íconos flotantes, luego, el sistema muestra una vista donde se visualizan los datos de la solicitud permitiendo ser modificados. Cuando se modifican los datos el sistema muestra el listado de solicitudes actualizado.	
Observaciones: Cuando una solicitud es modificada obtiene como estado pendiente.	
Prototipo de interfaz: Anexo 7	

Tabla 9: HU Modificar solicitud

2.6.1.7 Historia de usuario Buscar persona

Historia de usuario
Nombre historia de usuario: Buscar persona

Número: 07	**Referencia:** 08,09
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 2
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 12 horas
Descripción: La HU se inicia al seleccionar la opción personal del bloque, operaciones administrativas El sistema muestra una vista donde se listan todas las personas registradas y permite realizar la búsqueda de una o varias personas en específico, mostrando un selector que permite filtrar por criterios (estado de la persona y categoría de persona), además de un campo donde se pueden introducir otros criterios de búsqueda de la persona registrada (nombre y carné de identidad). Cuando se pulsa el botón buscar el sistema muestra un listado con todas las personas que cumplen con los criterios establecidos por el usuario.	
Observaciones:	
Prototipo de interfaz: Anexo 8 y Anexo 9	

Tabla 10: HU Buscar persona

2.6.1.8 Historia de usuario Modificar datos de persona

Historia de usuario	
Nombre historia de usuario: Modificar datos de persona	
Número: 8	**Referencia:** 10
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 2
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 10 horas
Descripción: La HU se inicia al concluir la búsqueda de la persona. Lo primero es seleccionar la persona en el listado y dar clic en el ícono que representa la opción modificar persona ubicado a la derecha del listado en la barra de íconos flotantes, luego, el sistema muestra una vista donde se visualizan los datos de la persona permitiendo ser modificados. Cuando se modifican los datos el sistema muestra el listado de personas actualizado.	
Observaciones:	

Prototipo de interfaz: Anexo 10

Tabla 11: HU Modificar datos de persona

2.6.1.9 Historia de usuario Dar baja

	Historia de usuario
Nombre historia de usuario: Dar baja	
Número: 09	**Referencia:** 12
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 2
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 6 horas
Descripción: La HU se inicia al concluir la búsqueda de la persona. Lo primero es seleccionar la persona en el listado y dar clic en el ícono que representa la opción dar baja ubicado a la derecha del listado en la barra de íconos flotantes, luego de pulsar este ícono el sistema actualiza el listado cambiando el estado de la persona.	
Observaciones:	
Prototipo de interfaz: Anexo 12	

Tabla 12: HU Dar baja

2.6.1.10 Historia de usuario Gestionar usuario

	Historia de usuario
Nombre historia de usuario: Gestionar usuario	
Número: 10	**Referencia:** 13,14,15,16,17
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 1
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 10 horas
Descripción: La HU se inicia cuando se selecciona la opción usuarios del bloque seguridad. Se muestra un listado de todos los usuarios existentes con una columna de opciones a la derecha, si el usuario selecciona la opción: **Crear usuario:** el sistema muestra una vista con los campos que deben ser llenados, luego	

de introducir los datos, se muestra un mensaje de confirmación “El usuario ha sido insertado” quedándose la vista activada brindando la posibilidad de crear otro usuario. En caso de que no se desee adicionar más usuarios se debe acceder a la opción listar, donde se muestra el listado de usuarios actualizado. **Modificar usuario:** luego de seleccionar el usuario a modificar, el sistema muestra una vista con los datos del usuario permitiendo su modificación, después de modificar los datos deseados el sistema muestra un mensaje de confirmación “El usuario ha sido modificado” y muestra el listado de usuarios con los datos actualizados. **Activar o Desactivar usuario:** luego de seleccionar el usuario y pulsar esta opción el sistema refresca la vista mostrando el usuario activado o desactivado de acuerdo con su estado inicial.
Observaciones: Para modificar un usuario primero se debe seleccionar el mismo en el listado de usuarios.
Prototipo de interfaz: Anexo 13

Tabla 13: HU Gestionar usuario

2.6.1.11 Historia de usuario Gestionar nomenclador rol

Historia de usuario	
Nombre historia de usuario: Gestionar nomenclador rol	
Número: 11	**Referencia:** 18,19,20,21,22
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 1
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 10 horas
Descripción: La HU se inicia cuando se selecciona la opción rol del bloque seguridad. Se muestra un listado de todos los roles existentes con una columna de opciones a la derecha, si el usuario selecciona la opción: **Crear rol:** el sistema muestra una vista con los campos que deben ser llenados, luego de introducir los datos, se muestra un mensaje de confirmación “El rol ha sido insertado” quedándose la vista activada brindando la posibilidad de crear otro rol. En caso de que no se desee adicionar más roles, se debe acceder a la opción listar donde se muestra el listado de	

roles actualizado. **Modificar rol:** luego de seleccionar el rol a modificar, el sistema muestra una vista con los datos del rol, permitiendo su modificación, después de modificar los datos deseados el sistema muestra un mensaje de confirmación “El rol ha sido modificado” y muestra el listado de roles con los datos actualizados. **Activar o Desactivar usuario:** luego de seleccionar el rol y pulsar esta opción el sistema refresca la vista mostrando el rol activado o desactivado de acuerdo con su estado inicial.
Observaciones: Para modificar un rol primero se debe seleccionar este en el listado de roles.
Prototipo de interfaz: Anexo 14

Tabla 14: HU Gestionar nomenclador rol

2.6.1.12 Historia de usuario Gestionar atributo de persona

Historia de usuario	
Nombre historia de usuario: Gestionar atributo de persona	
Número: 12	**Referencia:** 23,24,25,26
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 1
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 10 horas
Descripción: La HU se inicia cuando se selecciona la opción atributos de persona del bloque configuración. Se muestra un listado de todos los atributos de persona existentes, con una columna de opciones a la derecha, si el usuario selecciona la opción: **Crear atributo:** el sistema muestra una vista con los campos que deben ser llenados, luego de introducir los datos, se muestra un mensaje de confirmación “El atributo ha sido insertado” quedándose la vista activada brindando la posibilidad de crear otro atributo. En caso de que no se desee adicionar más atributos, se debe acceder a la opción listar, se muestra el listado de atributos de persona actualizado. **Modificar atributo:** luego de seleccionar el atributo a modificar, el sistema muestra una vista con los datos del atributo, permitiendo su modificación, después de modificar los datos	

deseados el sistema muestra un mensaje de confirmación “El atributo ha sido modificado” y muestra el listado de roles con los datos actualizados. **Activar o Desactivar atributo:** luego de seleccionar el atributo y pulsar esta opción el sistema refresca la vista mostrando al atributo activado o desactivado de acuerdo con su estado inicial.
Observaciones: Para modificar un atributo primero se debe seleccionar este en el listado de atributos.
Prototipo de interfaz: Anexo 15

Tabla 15: HU Gestionar atributo de persona

2.6.1.13 Historia de usuario Gestionar nomenclador área tercera

Historia de usuario	
Nombre historia de usuario: Gestionar nomenclador área tercera	
Número: 13	**Referencia:** 27,28,29,30,31
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 1
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 10 horas
Descripción: La HU se inicia cuando se selecciona la opción áreas terceras del bloque configuración. Se muestra un listado de todas las áreas terceras existentes, con una columna de opciones a la derecha, si el usuario selecciona la opción: **Crear área tercera:** el sistema muestra una vista con los campos que deben ser llenados, luego de introducir los datos, se muestra un mensaje de confirmación “El área ha sido insertada” quedándose la vista activada brindando la posibilidad de crear otra área. En caso de que no se desee adicionar más áreas tercera, se debe acceder a la opción listar, donde se muestra el listado actualizado de las áreas terceras. **Modificar área tercera:** luego de seleccionar el área a modificar, el sistema muestra una vista con los datos del área, permitiendo su modificación, después de modificar los datos deseados el sistema muestra un mensaje de confirmación “El área ha sido modificada” y muestra el listado de áreas con los datos actualizados.	

Activar o Desactivar área tercera: luego de seleccionar el área y pulsar esta opción el sistema refresca la vista mostrando el área activada o desactivada de acuerdo con su estado inicial.
Observaciones: Para modificar un área tercera primero se debe seleccionar esta en el listado de áreas terceras.
Prototipo de interfaz: Anexo 16

Tabla 16: HU Gestionar nomenclador área tercera

2.6.1.14 Historia de usuario Gestionar nomenclador categoría de persona

Historia de usuario	
Nombre historia de usuario: Gestionar nomenclador categoría de persona	
Número: 14	**Referencia:** 32,33,34,35,36
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 1
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 10 horas
Descripción: La HU se inicia cuando se selecciona la opción categorías de persona del bloque configuración. Se muestra un listado de todas las categorías de persona existentes, con una columna de opciones a la derecha, si el usuario selecciona la opción: **Crear categoría de persona:** el sistema muestra una vista con los campos que deben ser llenados, luego de introducir los datos, se muestra un mensaje de confirmación “La categoría de persona ha sido insertada” quedándose la vista activada brindando la posibilidad de crear otra categoría. En caso de que no se desee adicionar más categorías, se debe acceder a la opción listar, donde se muestra el listado de las categorías de persona actualizado. **Modificar categoría de persona:** luego de seleccionar la categoría a modificar, el sistema muestra una vista con los datos de la categoría permitiendo su modificación, después de modificar los datos deseados el sistema muestra un mensaje de confirmación “La categoría de persona ha sido modificada” y muestra el listado de categorías con los datos actualizados. **Activar o Desactivar categoría de persona:** luego de seleccionar la categoría y pulsar esta opción el sistema refresca la vista mostrando la categoría activada o desactivada de acuerdo con su estado inicial.	

Observaciones: Para modificar una categoría primero se debe seleccionar esta en el listado de categorías de persona.
Prototipo de interfaz: Anexo 17

Tabla 17: HU Gestionar nomenclador categoría de persona

2.6.1.15 Historia de usuario Obtener solicitudes por fecha

Historia de usuario	
Nombre historia de usuario: Obtener solicitudes por fecha	
Número: 15	**Referencia:** 37
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 3
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 14 horas
Descripción: La HU permite obtener un listado con todas las solicitudes realizadas dado una fecha determinada por el usuario.	
Observaciones:	
Prototipo de interfaz: Anexo 18	

Tabla 18: HU Obtener solicitudes por fecha

2.6.1.16 Historia de usuario Obtener personal por estado de persona

Historia de usuario	
Nombre historia de usuario: Obtener personal por estado de persona	
Número: 16	**Referencia:** 38
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 3
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 14 horas
Descripción: La HU permite obtener todo el personal existente por estado de persona (de alta o de baja).	
Observaciones:	
Prototipo de interfaz: Anexo 19	

Tabla 19: HU Obtener personal por estado de persona

2.6.1.17 Historia de usuario Obtener personal por categoría de persona

Historia de usuario	
Nombre historia de usuario: Obtener personal por categoría de persona	
Número: 17	**Referencia:** 39
Programadores: Yunet Labrada y Mayelin Illana	**Iteración Asignada:** 3
Prioridad: Alta	**Puntos Estimados:** 12 horas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 14 horas
Descripción: La HU permite obtener un listado de todo el personal existente de acuerdo con la categoría de persona.	
Observaciones:	
Prototipo de interfaz: Anexo 20	

Tabla 20: HU Obtener personal por categoría de persona

Conclusiones

En este capítulo, a partir de la concepción inicial del sistema y todas las funcionalidades identificadas por el usuario, se define la LRP, así como la prioridad de cada una de estas, las cuales quedan recogidas en HU, donde se describen los requisitos del sistema. La planificación de iteraciones se realizó teniendo en cuenta la prioridad para el negocio de cada historia de usuario. Dada la propuesta de solución planteada se puede comenzar a implementar el Sistema de Registro de Personal UCI.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN

En el presente capítulo se modelan los artefactos que exige la metodología ágil SCRUM_XP para la fase de diseño e implementación. Se describe como funciona la arquitectura del sistema y los servicios web que se consumen de otros sistemas ya existentes en la universidad. Se presenta el modelo de datos y el de despliegue, se describen las tareas de ingeniería y el estándar de codificación.

3.1 Descripción de la arquitectura

El sistema propuesto cuenta una arquitectura basada en uno de los estilos más utilizados para el diseño web, el patrón MVC, explicado en el Capítulo 1. Este patrón le brinda al sistema una clara definición de como es mostrada la información, el manejo de las acciones que el usuario desea hacer sobre él y como se realizan estas acciones modificando y validando la información.

La estructura del patrón MVC propuesto por el framework utilizado CodeIgniter, se evidencia de la siguiente forma:

La vista cuenta con dos elementos fundamentales: una plantilla que es el archivo que contiene el código HTML que no varía para las interfaces de usuario, un área de trabajo que contiene los formularios de entrada y las vistas de salida de datos. En el controlador se encuentran las clases controladoras que son las encargadas de manejar los eventos del sistema. Las librerías son clases en las cuales se desarrolla la lógica del negocio. La clase administradora se encarga de gestionar dentro de todas las modelos existentes, cual es la modelo que la librería está invocando. En la modelo se encuentra la capa de acceso a datos que contiene todas las funcionalidades que responden a la lógica del negocio.

De esta forma cuando el usuario interactúa con la aplicación lo hace mediante la vista, que envía las peticiones al controlador, este envía el flujo de datos hacia las librerías y esta hacia la acción correspondiente que sería la clase administradora la cual se encarga de llamar a la función implementada en la modelo, esta recepciona el resultado emitido por la modelo y se la envía a la librería, esta transforma la información y se la devuelve a la controladora la cual levanta una vista donde se muestran los resultados solicitados por el usuario.

3.2 Estrategia de integración

El sistema tiene la particularidad de consumir información gestionada por otros sistemas ya existentes en la universidad. Esta comunicación se realiza mediante el protocolo SOAP para la comunicación con los servicios web, a continuación se especifican las diferentes funcionalidades de las cuales se obtiene dicha información.

ASSETS (Sistema de Registro Integral)

- Obtener cargos: este servicio permite obtener un arreglo que contiene todos los cargos que pueden ser desempeñados en un área de trabajo, según los cargos oficiales que emite el Ministerio de Trabajo y Seguridad Social.
- Obtener áreas: permite obtener las diferentes áreas que existen en la universidad.
- Obtener cargo dado un identificador: a este servicio recibe como parámetro un identificador y devuelve el nombre del cargo asociado a dicho identificador.
- Obtener área dado identificador: permite obtener dado un identificador pasado por parámetro el nombre del área correspondiente.

RESIDENCIA

- Obtener edificios: este servicio devuelve todos los edificios que existes en la universidad.

LDAP (Protocolo Ligero de Acceso a Directorios)

- Comprobar usuario: este servicio realiza una búsqueda interna para comprobar si el usuario recibido por parámetro es válido en el directorio activo de la universidad.

3.3 Modelo de datos

Para la persistencia de la información necesaria y para el correcto funcionamiento del sistema, se identificaron 18 tablas. Ver **figura 3**

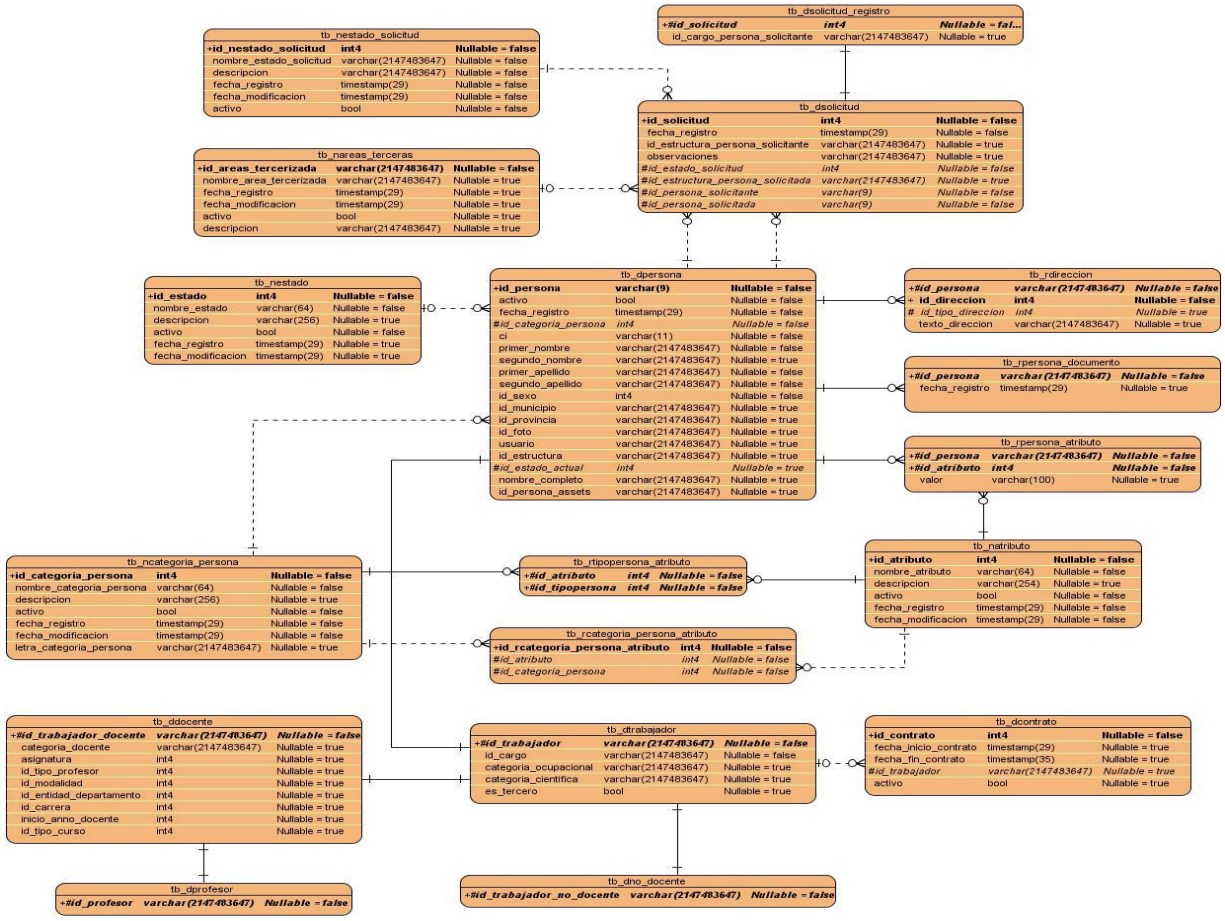


Figura 3: Modelo de datos

A continuación se explican las principales tablas:

tb_dpersona: entidad que almacena los datos base de una persona.

tb_rpersona_atributo: entidad que establece la relación entre los atributos asociados a una persona.

tb_natributo: entidad que almacena los posibles atributos de un tipo de persona.

tb_rcategoria_persona_atributo: entidad que establece la relación entre los atributos correspondientes a las distintas categorías de persona.

tb_ncategoria_persona: entidad que almacena las diferentes categorías de persona.

tb_nestado: entidad que almacena los diferentes estados.

tb_dsolicitud: entidad que almacena las solicitudes realizadas.

tb_dtrabajador: entidad que almacena los datos base de un trabajador.

tb_ddocente: entidad que almacena los datos base de un trabajador docente.

tb_dno_docente: entidad que almacena los datos de un trabajador no docente.

tb_dcontrato: entidad que almacena la fecha en que se realizan los contratos.

tb_dno_trabajador: entidad que almacena los datos base de un personal no trabajador.

tb_dfamiliar: entidad que almacena los datos de un familiar.

tb_dvisitas: entidad que almacena los datos de un visitante.

tb_dprofesor: entidad que almacena los datos de un profesor.

tb_dsolicitud_registro: entidad que almacena el cargo de las personas que realizan la solicitud.

tb_nestado_solicitud: entidad que almacena los diferentes estados que puede tener una solicitud.

tb_nareas_tercerras: entidad que almacena las diferentes áreas terceras que existen en la universidad.

3.4 Tareas generales de la implementación (TI)

Para la implementación del sistema se llevan a cabo una serie de tareas que no se encuentran comprendidas en las historias de usuarios y que se han definido como tareas generales a realizar. Las tareas de la implementación (tareas de ingeniería) son muy importantes para el programador porque guían el proceso de desarrollo del sistema.

	Tarea de ingeniería
Número de la tarea: 01	**Historias de usuario implicadas:**

Nombre de la tarea: Diseñar base de datos	
Tipo de tarea: Desarrollo	**Puntos estimados:** 0.1
Fecha de inicio: 28/10/2009	**Fecha de fin:** 12/4/2010
Programador responsable: Mayelin Illana y Yunet Labrada	
Descripción: Utilizando PostgreSQL como gestor, se crea la base de datos en la cual se almacenarán los datos persistentes generados durante la utilización de la aplicación.	

Tabla 24: TI Diseñar base de datos

		Tarea de ingeniería
Número de la tarea: 02	**Historias de usuario implicadas:**	
Nombre de la tarea: Generar las clases modelo		
Tipo de tarea: Desarrollo		**Puntos estimados:** 0.1
Fecha de inicio: 10/4/2010		**Fecha de fin:** 4/6/2010
Programador responsable: Mayelin Illana y Yunet Labrada		
Descripción: Se generan las clases modelo utilizando el script *php_gen.php*, especificándole el servidor de bases de datos, el nombre de la base de datos, el puerto a usar, el usuario y la contraseña para la conexión.		

Tabla 25: TI Generar las clases modelo

		Tarea de ingeniería
Número de la tarea: 03	**Historias de usuario implicadas:**	
Nombre de la tarea: Implementar la clase administradora		
Tipo de tarea: Desarrollo		**Puntos estimados:** 1
Fecha de inicio: 10/4/2010		**Fecha de fin:** 4/6/2010
Programador responsable: Mayelin Illana y Yunet Labrada		
Descripción: Se implementa la clase *registro_mng* que agrupa por funcionalidad las clases modelo.		

Tabla 26: TI Implementar la clase administradora

	Tarea de ingeniería
Número de la tarea: 04	**Historias de usuario implicadas:**
Nombre de la tarea: Confeccionar las interfaces de usuario	
Tipo de tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 10/4/2010	**Fecha de fin:** 4/6/2010
Programador responsable: Mayelin Illana y Yunet Labrada	
Descripción: Se diseñan todas las interfaces de usuario que tendrá el sistema de acuerdo con las funcionales planteadas en LRP.	

Tabla 27: TI Confeccionar las interfaces de usuario

	Tarea de ingeniería
Número de la tarea: 05	**Historias de usuario implicadas:** 1
Nombre de la tarea: Implementar los métodos en las clases controladoras	
Tipo de tarea: Desarrollo	**Puntos estimados:**
Fecha de inicio: 10/4/2010	**Fecha de fin:** 4/6/2010
Programador responsable: Mayelin Illana y Yunet Labrada	
Descripción: Se implementan en las clases controladoras los métodos con sus validaciones.	

Tabla 28: TI Implementar los métodos en las clases controladoras

	Tarea de ingeniería
Número de la tarea: 06	**Historias de usuario implicadas:**
Nombre de la tarea: Implementar métodos en las librerías	
Tipo de tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 10/4/2010	**Fecha de fin:** 4/6/2010
Programador responsable: Mayelin Illana y Yunet Labrada	

Descripción: Se implementan los métodos en las librerías correspondientes, que reciben los datos y se hace una llamada al método correspondiente de la clase modelo desde la clase administradora, pasándole dichos valores.

Tabla 29: TI Implementar métodos en las librerías

3.5 Estándar de codificación

3.5.1 Identación, llaves de apertura, de cierre y tamaño de las líneas

Usar una identación sin tabulaciones, con un equivalente a cuatro espacios, el uso de las llaves "{}" será en una nueva línea. La longitud de las líneas de código es aproximadamente de 75-80 caracteres, para mantener la legibilidad del código.

3.5.2 Conversión de nomenclatura

Variables: se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

Constantes: siempre debe ser todas en mayúsculas, con caracteres de subrayado "_" para separar palabras en caso de nombres compuestos.

Clases: siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con el carácter subrayado "_" y el resto en minúscula.

Funciones: se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa.

Ficheros: todo siempre en minúscula y en caso de nombres compuestos se usa el carácter subrayado "_".

1. **Vistas:** intuitivo y relacionado con el formulario y/o vista que representa.
2. **Modelos:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo _mdl o _base en caso de ser modelos base.

3. **Librerías:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo _lib.
4. **Controladoras:** con el mismo nombre de la clase que representa.
5. **Manager:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo _mng.

3.5.3 Estructuras de control

Se incluyen las palabras reservadas tales como: if, for, foreach, while, switch, entre las estructuras de control y entre los paréntesis debe de existir un espacio. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales, esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos.

Si las condiciones son muy largas que sobrepasan el tamaño de la línea, estas se dividen en varias líneas. En el mejor de los casos cuando la condición es muy extensa, se puede dividir esta en variables y compararlas dentro de la estructura de control.

3.6 Servicios brindados

- Obtener personal de alta
- Obtener personal de baja
- Obtener categorías de persona
- Obtener áreas terceras
- Obtener personas

3.7 Diagrama de despliegue

Un diagrama de despliegue muestra como y donde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos, y la construcción interna puede ser representada por nodos o artefactos embebidos. Como los artefactos se ubican en los nodos para modelar el despliegue del sistema, la ubicación es guiada por el uso de las especificaciones de despliegue. Ver **figura 4**

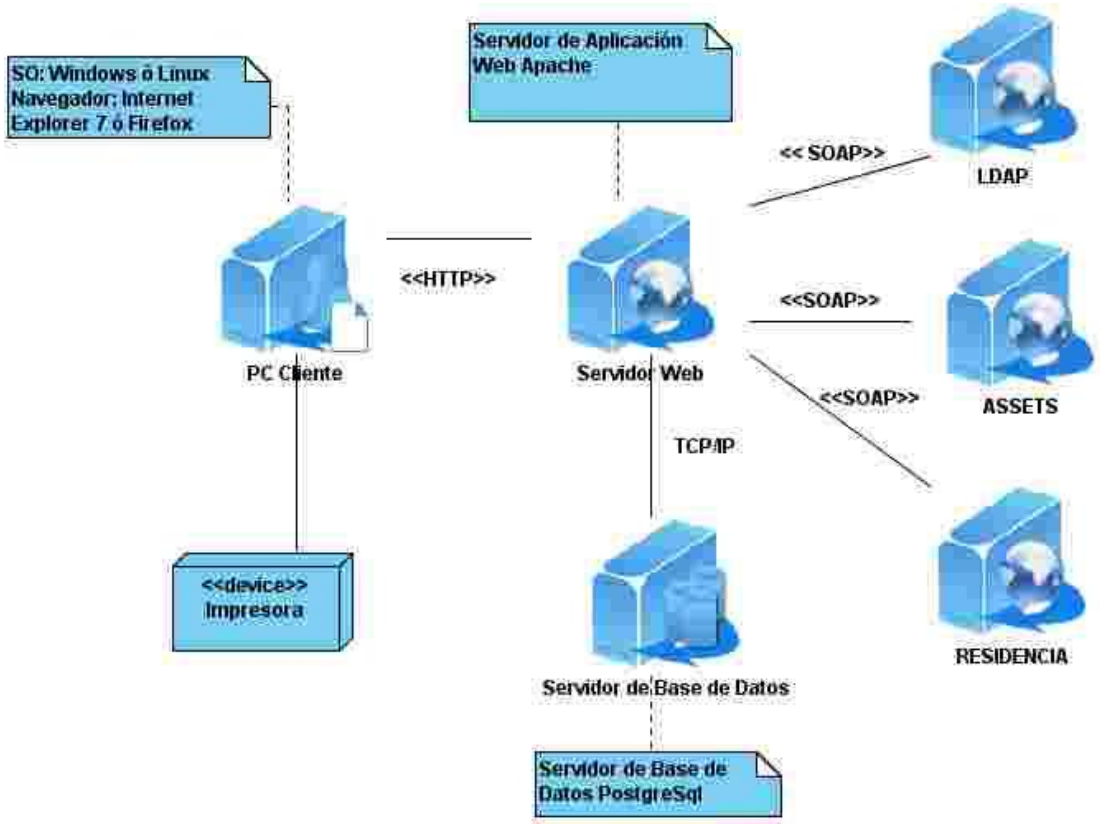


Figura 4: Diagrama de despliegue

Conclusiones

En este capítulo se realizó la presentación y descripción de la arquitectura propuesta para la construcción del sistema. Se definió el diseño de la base de datos, la descripción de las tareas de ingeniería y los estándares de codificación que brindan una idea más clara de como desarrollar una aplicación web eficiente y confiable. Se especificaron los servicios web brindados y consumidos por otros sistemas existentes en la universidad y finalmente se mostró el despliegue de la aplicación donde se evidencia la configuración, sobre la cual deberá ser implantado el Sistema de Registro y Control de Personal para su correcto funcionamiento.

CONCLUSIONES

Con el desarrollo de la presente investigación se analizaron los procesos de registro del personal tercero, familiar y visitante en la UCI. Se determinó la necesidad de desarrollar un sistema informático, capaz de solucionar todos los problemas existes en la universidad, en cuanto al registro de estas personas. El desarrollo de las tareas de la investigación posibilitó dar cumplimiento al objetivo general planteado, arribando a las siguientes conclusiones:

- Los sistemas estudiados no satisfacen las necesidades, que en materia de gestión de la información tiene el registro de personal en la universidad.
- Se logró obtener un sistema que integra el registro de visitantes, familiares y terceros de la Universidad de las Ciencias Informáticas, cumpliendo con las funcionalidades previstas y teniendo en cuenta las normas técnicas y las tecnologías aprobadas por el Centro de Informatización Universitaria para el desarrollo de sistemas.
- El sistema facilitó la obtención de diferentes reportes necesarios para los clientes.
- Se desarrolló un servicio web con el objetivo de brindarle información a los sistemas de identificación y la intranet 2.0.

RECOMENDACIONES

Luego de culminada la investigación y cumplido el objetivo general de la misma, en vista a mejorar el funcionamiento del sistema y con el fin de que se incrementen las funcionalidades brindadas para un mejor uso de este, se recomienda:

- Desarrollar un manual de usuario para uso de los clientes, con el objetivo de lograr un mayor entendimiento del sistema y su funcionamiento.
- Al equipo de proyecto enriquecer las diferentes salidas estadísticas y reportes asociados, haciendo uso de graficas dinámicas, para permitirle a la capa de gerencia un mejor entendimiento en la toma de decisiones.
- Al equipo de proyecto extender la seguridad del sistema hasta el nivel de funcionalidad, para lograr un sistema robusto y poco vulnerable.

REFERENCIAS BIBLIOGRÁFICAS

[1] Control de Persona y Recursos Humanos. [En línea] [Citado: enero 10, 2010.] http://www.elprisma.com/apuntes/administracion_de_empresas/controldepersonal/.

[2] Empleados MSD Multiusuario. [En línea] [Citado: febrero 10, 2010.] http://www.abcdatos.com/programas/programa/z6106.html.

[3] Wikimetal RRHHv3.0.[En línea] [Citado: febrero 3, 2010.] http://www.abcdatos.com/programas/programa/z7511.html.

[4] González Acosta, Roberto. SISTEMA PARA LA GESTIÓN DE LOS COLABORADORES DE LA SALUD.

[5] APLICACIÓN INFORMÁTICA PARA EL CONTROL DE LOS RECURSOS HUMANOS EN EL SISTEMA NACIONAL DE SALUD. [En línea] [Citado: abril 12, 2010.]

[6] Selsof. [En línea] [Citado: marzo 2, 2010.] http://www.desoft.cu/Productos1/Selsoft/tabid/435/Default.aspx.

[7] Fastos. [En línea] [Citado: febrero15,2010.] http://www.desoft.cu/Productos1/FastosPagus/tabid/442/Default.aspx.

[8] Arquitectura Cliente/Servidor. [En línea] [Citado: enero 23, 2010.] http://www.csi.map.es/csi/silice/Global71.html.

[9] ¿Qué es Javascript?.[En línea] [Citado: febrero 22,2010.] http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/.

[10] Lenguaje de programación para páginas web. [En línea] [Citado: junio 8, 2010.] http://www.monografias.com/trabajos7/html/html.shtml#crea.

[11] Definición de CSS. [En línea] [Citado: junio 8, 2010.] http://www.masadelante.com/faqs/css.

[12] Ajax: Un nuevo acercamiento a las aplicaciones web. [En línea] [Citado: enero 22, 2010.] http://www.maestrosdelweb.com/editorial/ajax/

[13] Un vistazo a PHP5. [En línea] [Citado: enero 23, 2010.] http://libertonia.escomposlinux.org/story/2004/7/15/115328/134.

[14] CodeIgniter, un gran framework para PHP. [En línea] [Citado: febrero 22, 2010.] http://www.joseduenas.com/archivos/codeigniter-un-gran-framework.

[15] Apache Software Foundation. [En línea] [Citado: febrero 02, 2010.] http://www.apache.org/.

[16] BPMN. [En línea] [Citado: febrero 15,2010.] http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm.

[17] Ingeniería de Software UML. [En línea] [Citado: febrero 21, 2010.] http://www.monografias.com/trabajos5/insof/insof.shtml.

[18] Visual Paradigm for UML. [En línea] [Citado: enero 21, 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/

[19] BizAgi Process Modeler. [En línea] [Citado: febrero 15, 2010.] http://www.softdownload.org/programas/bizagi-process-modeler-1-5.asp.

[20] Sección de ems postgresql. [En línea] [Citado: junio 7, 2010.] http://www.freedownloadmanager.org/es/downloads/ems_postgresql_libre.

BIBLIOGRAFÍA

¿Qué es Javascript?. [En línea] [Citado: febrero 22,2010.] http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/.

¿Qué es un Servidor Web?. Linux, Java y programación. [En línea] [Citado: febrero 21, 2010.] http://www.misrespuestas.com/que-es-un-servidor-web.html.

¿Qué es PHP?. [En línea] [Citado: enero 22, 2010.] http://www.desarrolloweb.com/articulos/392.php.

AJAX un nuevo acercamiento a aplicaciones Web. [En línea] [Citado: marzo 04, 2010.] http://www.maestrosdelweb.com/editorial/ajax.

Ajax: Un nuevo acercamiento a las aplicaciones web. [En línea] [Citado: febrero 15, 2010.] http://www.maestrosdelweb.com/editorial/ajax/.

Álvaro del Castillo, Félix. El servidor de web Apache: Introducción práctica. Apache 1.x y 2.0 alpha. [En línea] [Citado: febrero 15,2010.] http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/apache.pdf.

Apache Software Foundation. [En línea] [Citado: febrero 02, 2010.] http://www.apache.org/.

APLICACIÓN INFORMÁTICA PARA EL CONTROL DE LOS RECURSOS HUMANOS EN EL SISTEMA NACIONAL DE SALUD. [En línea] [Citado: abril 12, 2010.]

Arquitectura Cliente/Servidor. [En línea] [Citado: enero 23, 2010.] http://www.csi.map.es/csi/silice/Global71.html.

BizAgi Process Modeler. [En línea] [Citado: febrero 15, 2010.] http://www.softdownload.org/programas/bizagi-process-modeler-1-5.asp.

BPMN. [En línea] [Citado: febrero 15,2010.] http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm.

Curso 2009-2010. Clases de Ingeniería del Software I. curso 2009-2010.

CodeIgniter,un gran framework para PHP. [En línea] [Citado: febrero 22, 2010.] http://www.joseduenas.com/archivos/codeigniter-un-gran-framework.

Control de Persona y Recursos Humanos. [En línea] [Citado: enero 10, 2010.] http://www.elprisma.com/apuntes/administracion_de_empresas/controldepersonal/.

Definición de CSS. [En línea] [Citado: junio 8, 2010.] http://www.masadelante.com/faqs/css.

Desarrollo Web. [En línea] [Citado: 1 22, 2010.] http://www.desarrolloweb.com/articulos/1622.php.

Empleados MSD Multiusuario. [En línea] [Citado: febrero 10, 2010.] http://www.abcdatos.com/programas/programa/z6106.html.

Fastos. [En línea] [Citado: febrero 15,2010.] http://www.desoft.cu/Productos1/FastosPagus/tabid/442/Default.aspx.

González Acosta, Roberto. SISTEMA PARA LA GESTIÓN DE LOS COLABORADORES DE LA SALUD.

Ingeniería de Software UML. [En línea] [Citado: febrero 21, 2010.] http://www.monografias.com/trabajos5/insof/insof.shtml.

Lenguaje de programación para paginas web. [En línea] [Citado: junio 8, 2010.] http://www.monografias.com/trabajos7/html/html.shtml#crea.

Manual JQuery. [En línea] [Citado: febrero 15, 2010.] http://www.desarrolloweb.com/manuales/manual-jquery.html.

Programación Web. [En línea] [Citado: febrero 20, 2010.] http://www.programacionweb.net/articulos/.

Sección de ems postgresql. [En línea] [Citado: junio 7, 2010.] http://www.freedownloadmanager.org/es/downloads/ems_postgresql_libre.

Selsof. [En línea] [Citado: marzo 2, 2010.] http://www.desoft.cu/Productos1/Selsoft/tabid/435/Default.aspx.

Tutorial de PostgreSQL. [En línea] [Citado: febrero 3, 2010.] http://sdi.bcn.cl/desarrollo/doctos/PostgreSQL%20-%20Tutorial.pdf.

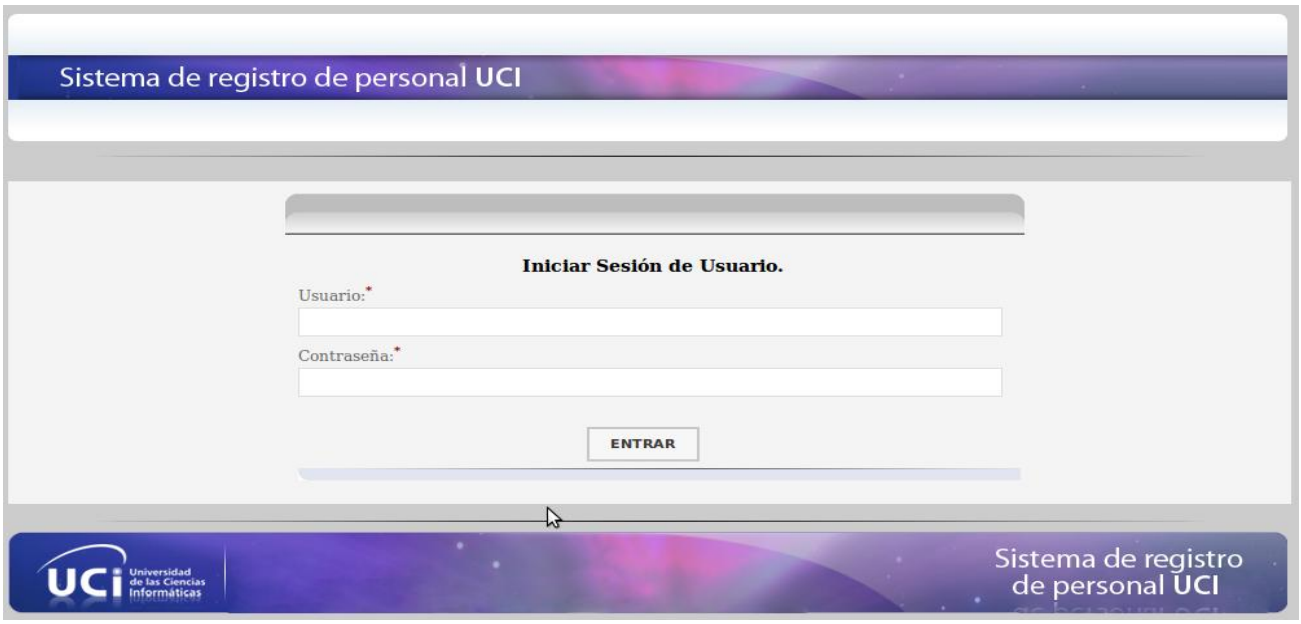
Tutorial de Scrum y XP. Tutorial de Scrum y XP. [En línea] [Citado: marzo 4, 2010.] http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf.

Un vistazo a PHP5. [En línea] [Citado: enero 23, 2010.] http://libertonia.escomposlinux.org/story/2004/7/15/115328/134.

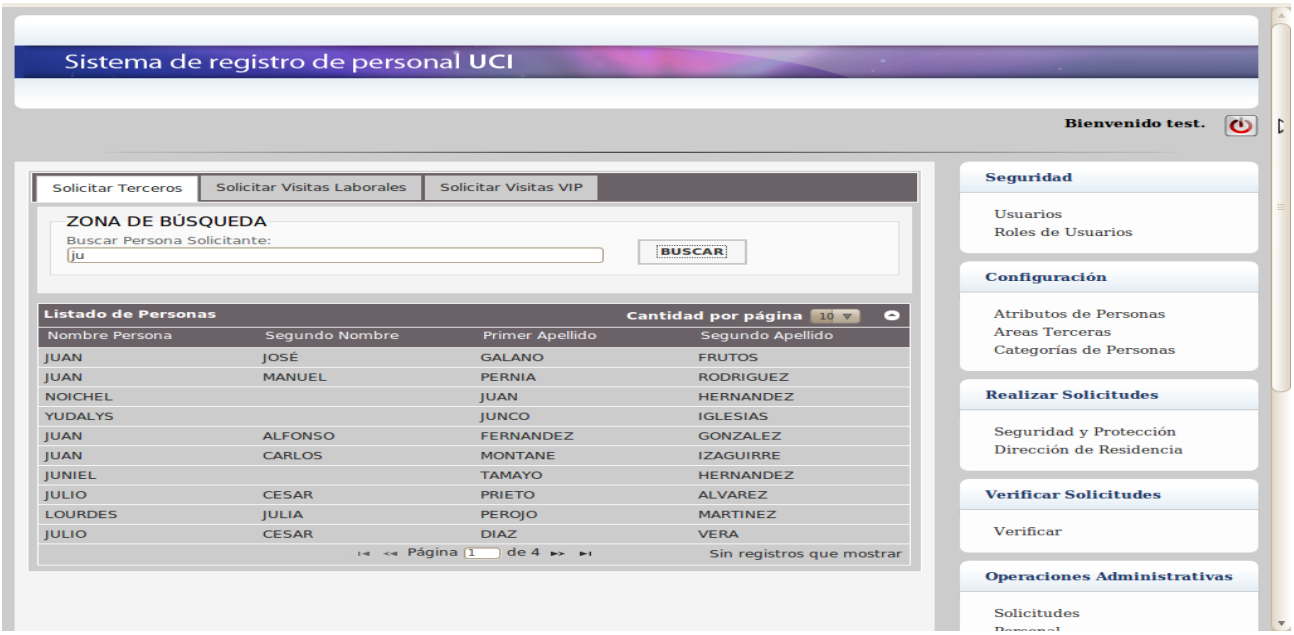
Visual Paradigm for UML. [En línea] [Citado: enero 21, 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/

Wikimetal RRHHv3.0. [En línea] [Citado: febrero 3,2010.] http://www.abcdatos.com/programas/programa/z7511.html.

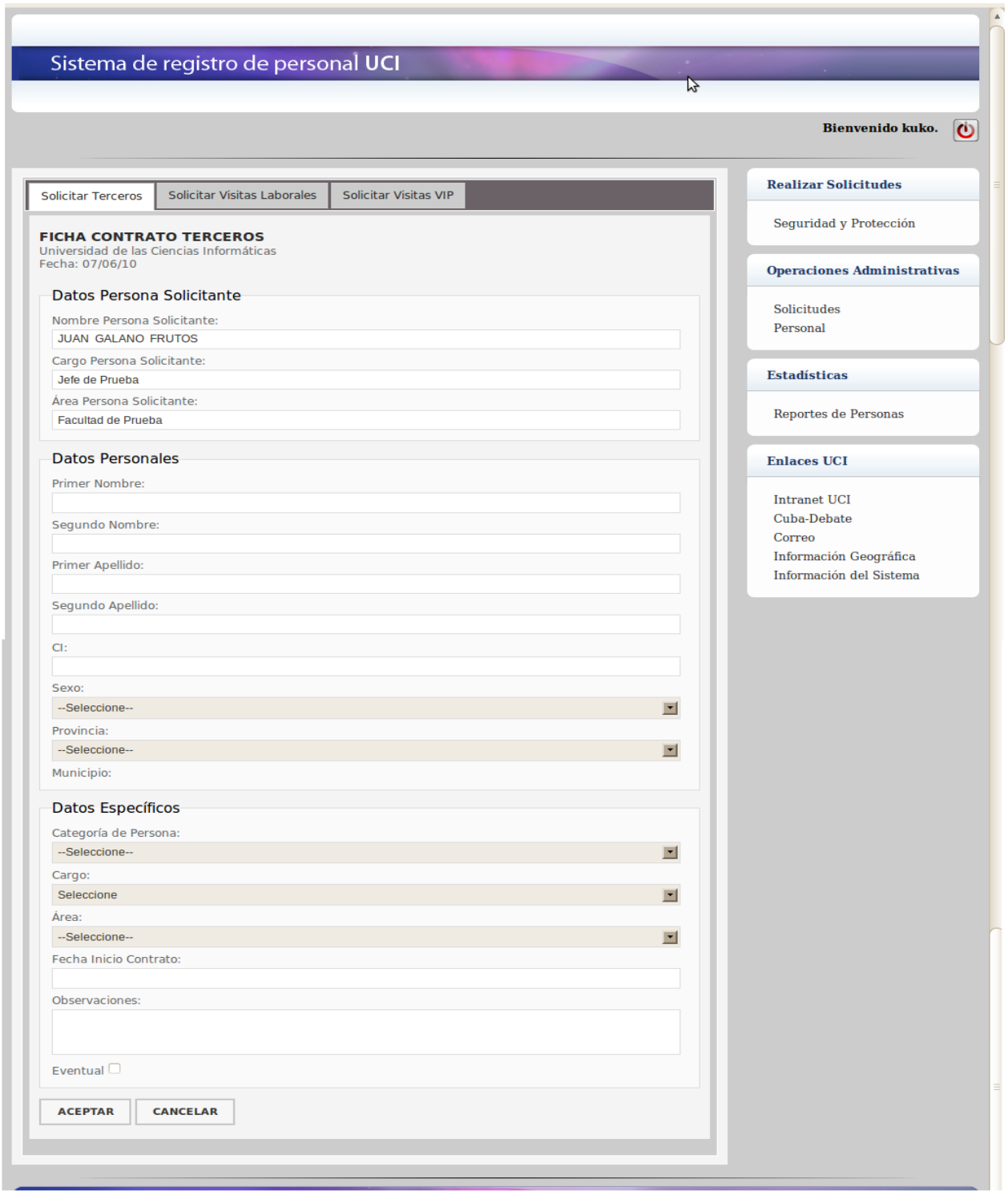
ANEXOS



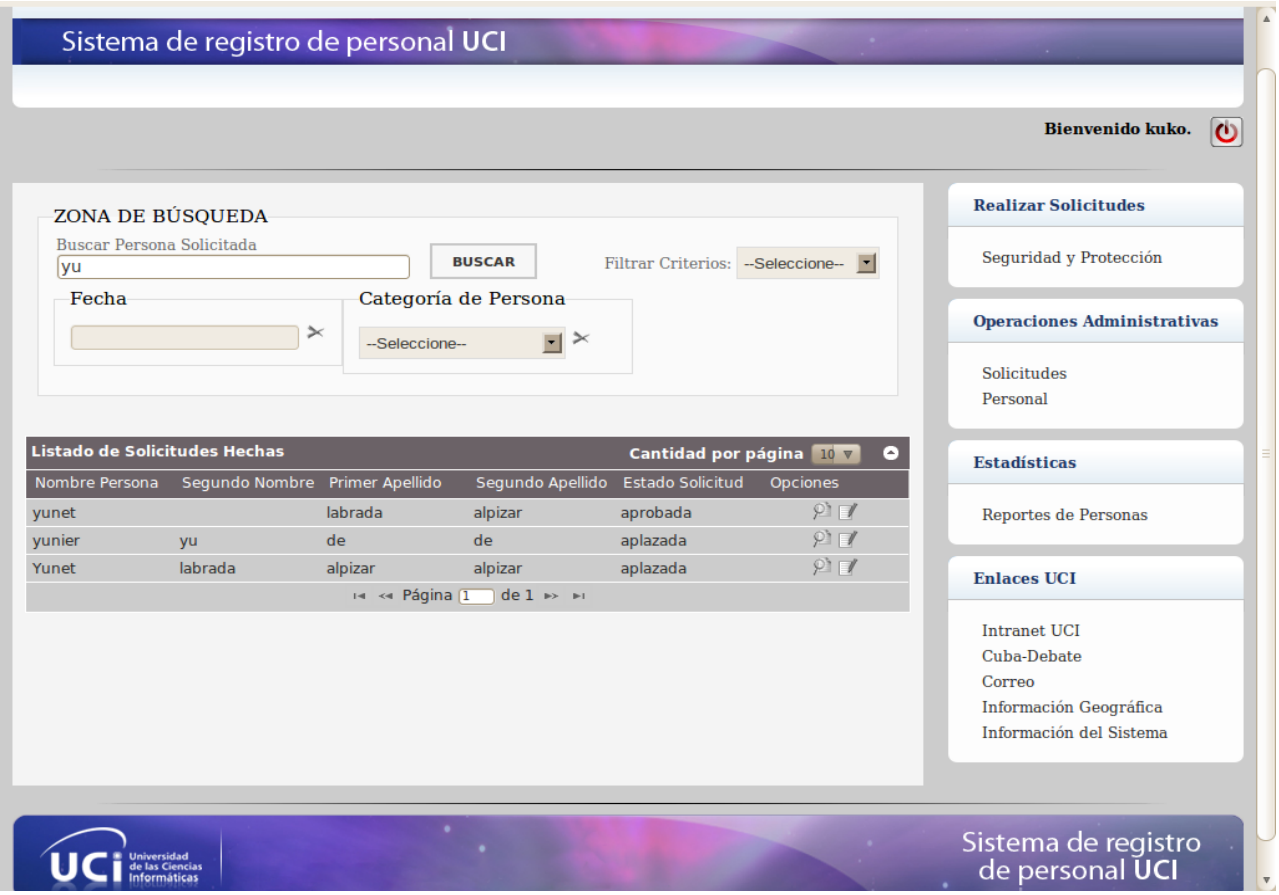
Anexo 1: Vista Autenticar usuario



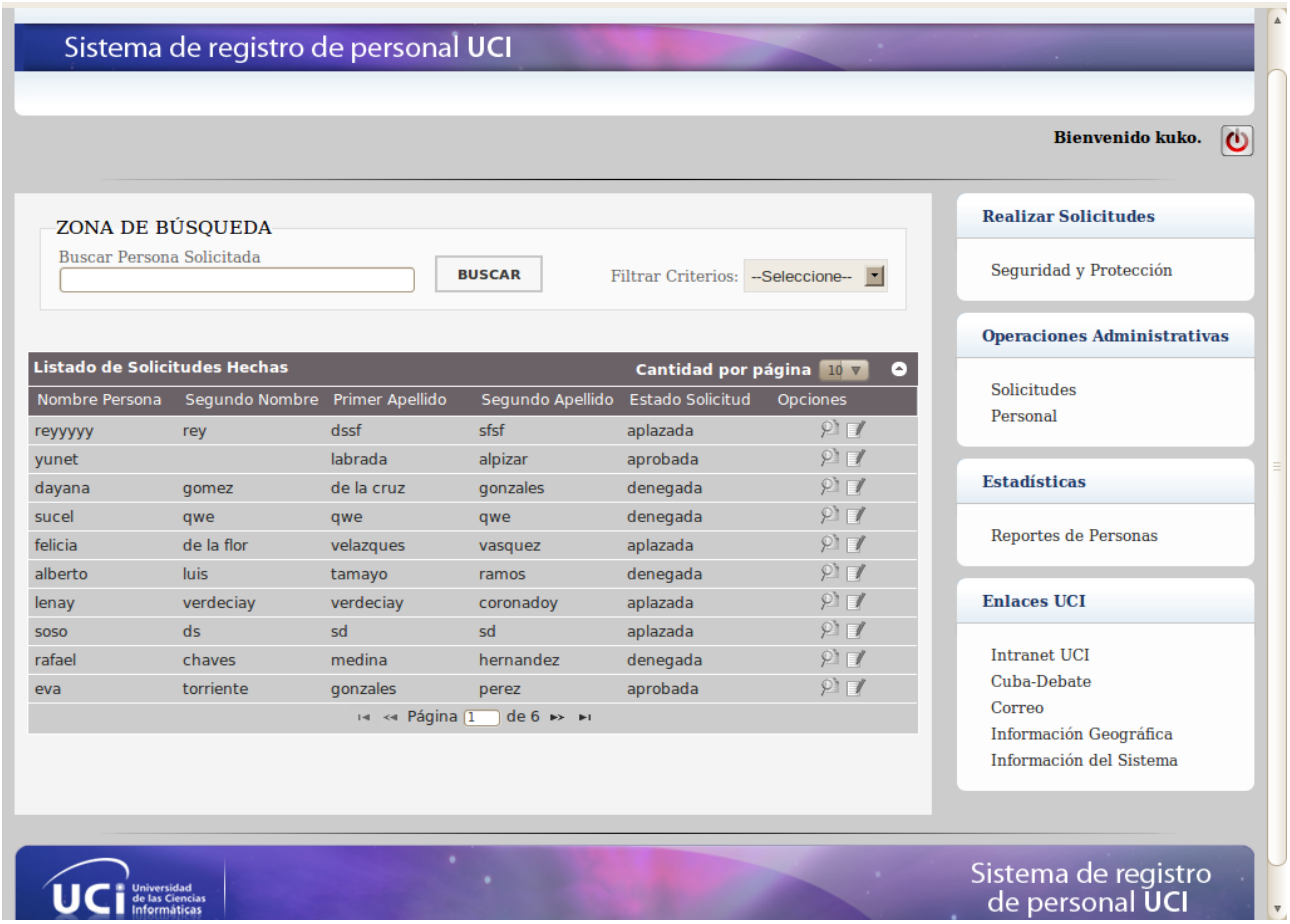
Anexo 2: Vista Buscar persona solicitante



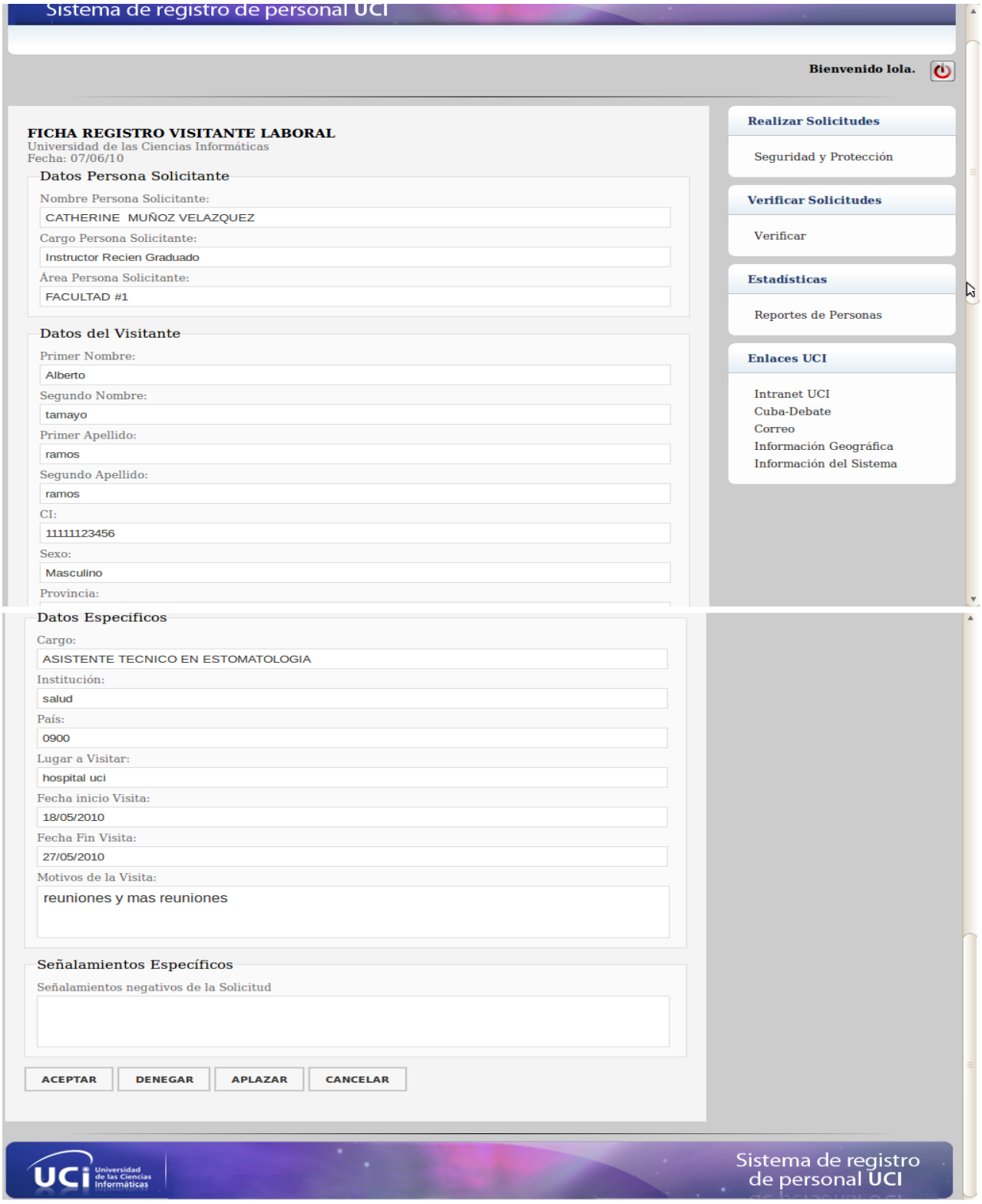
Anexo 3: Vista Realizar solicitud



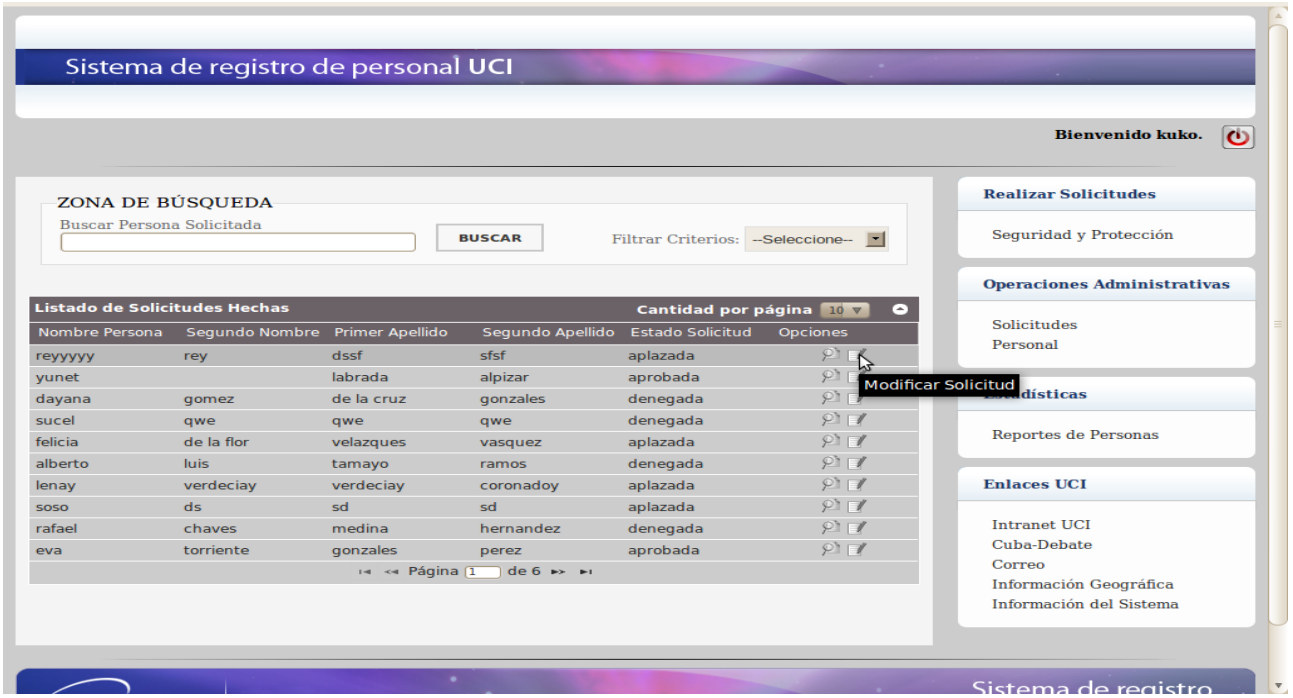
Anexo 4: Vista Buscar solicitud



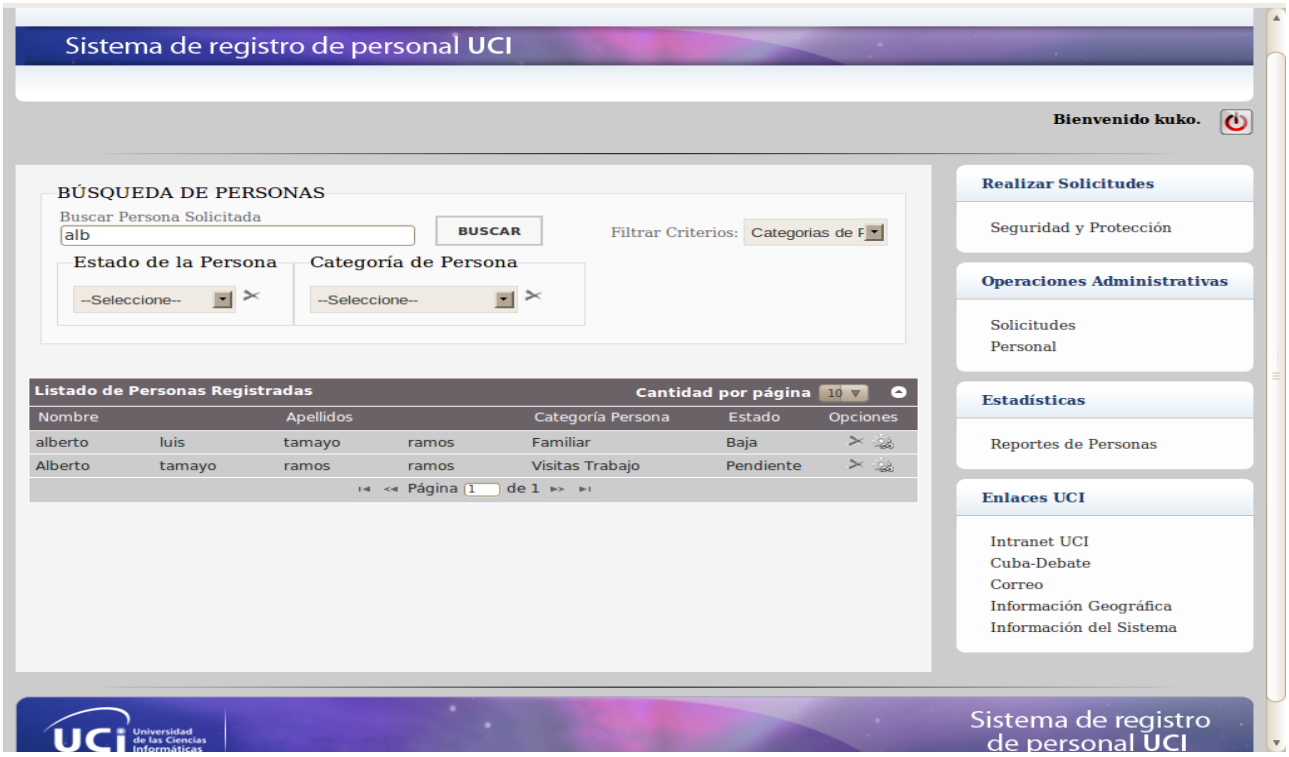
Anexo 5: Vista Listar solicitudes



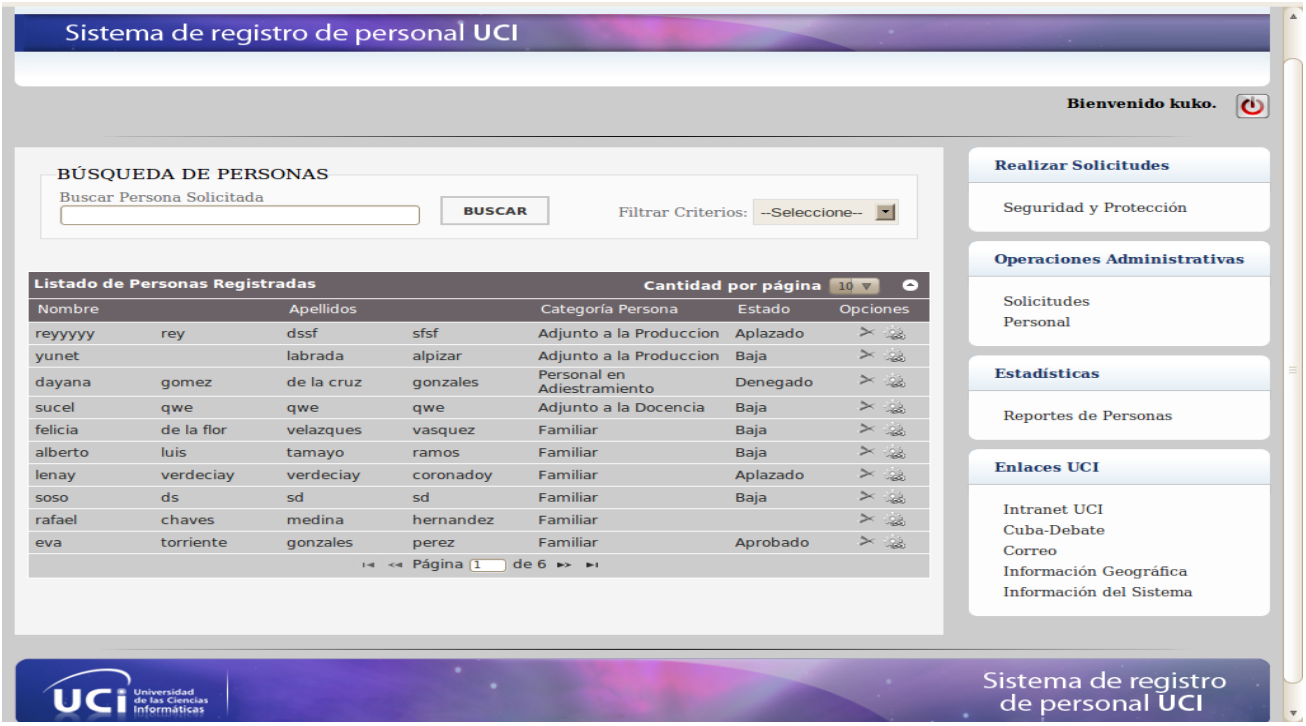
Anexo 6: Vista Verificar solicitud



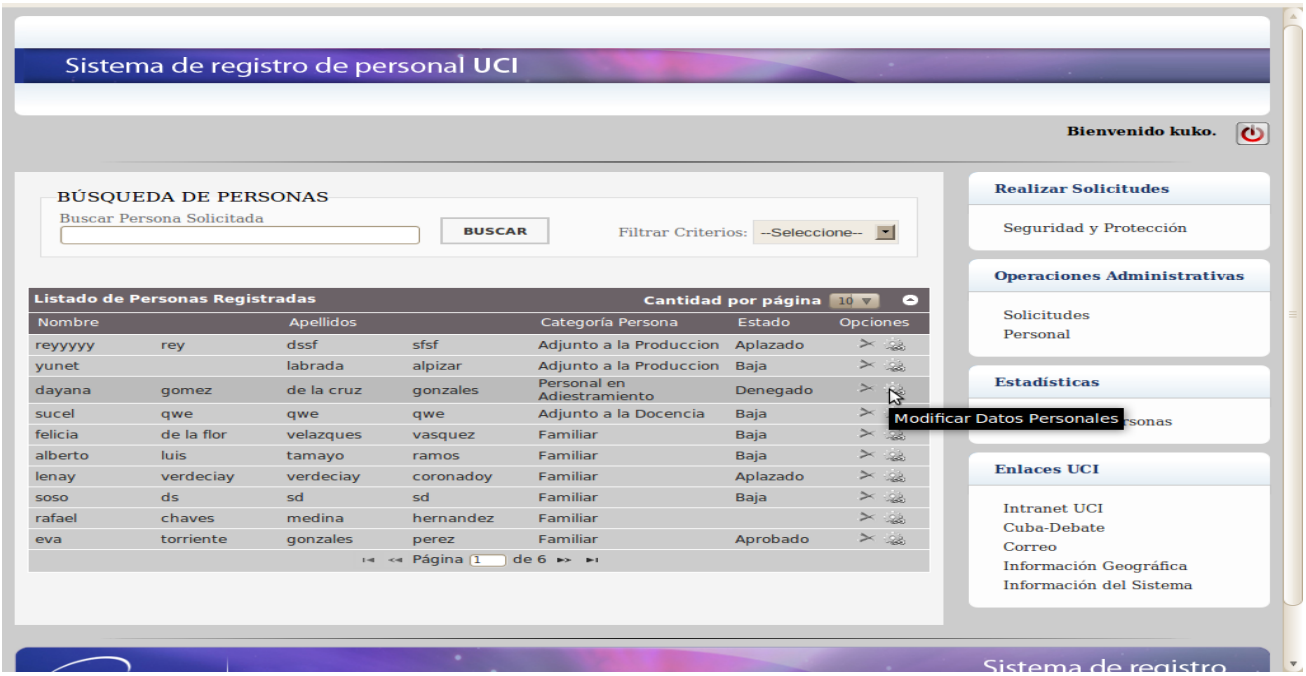
Anexo 7: Vista Modificar solicitud



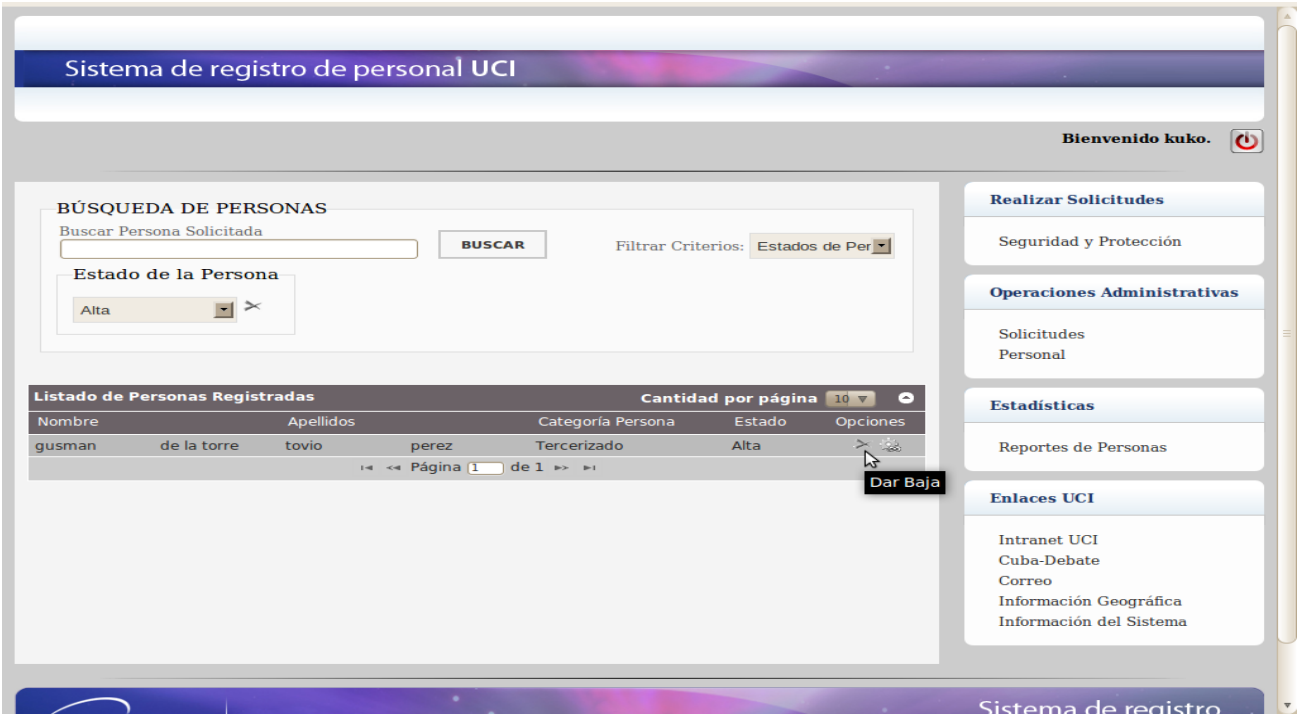
Anexo 8: Vista Buscar persona



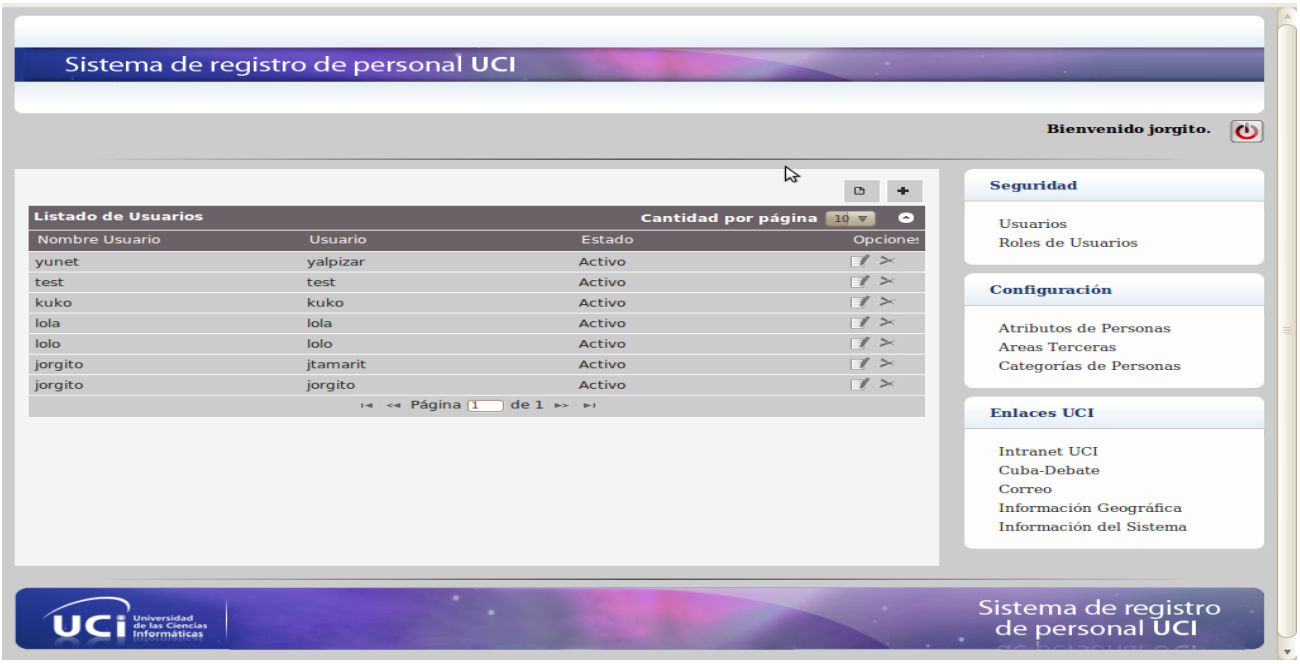
Anexo 9: Vista Listar personas



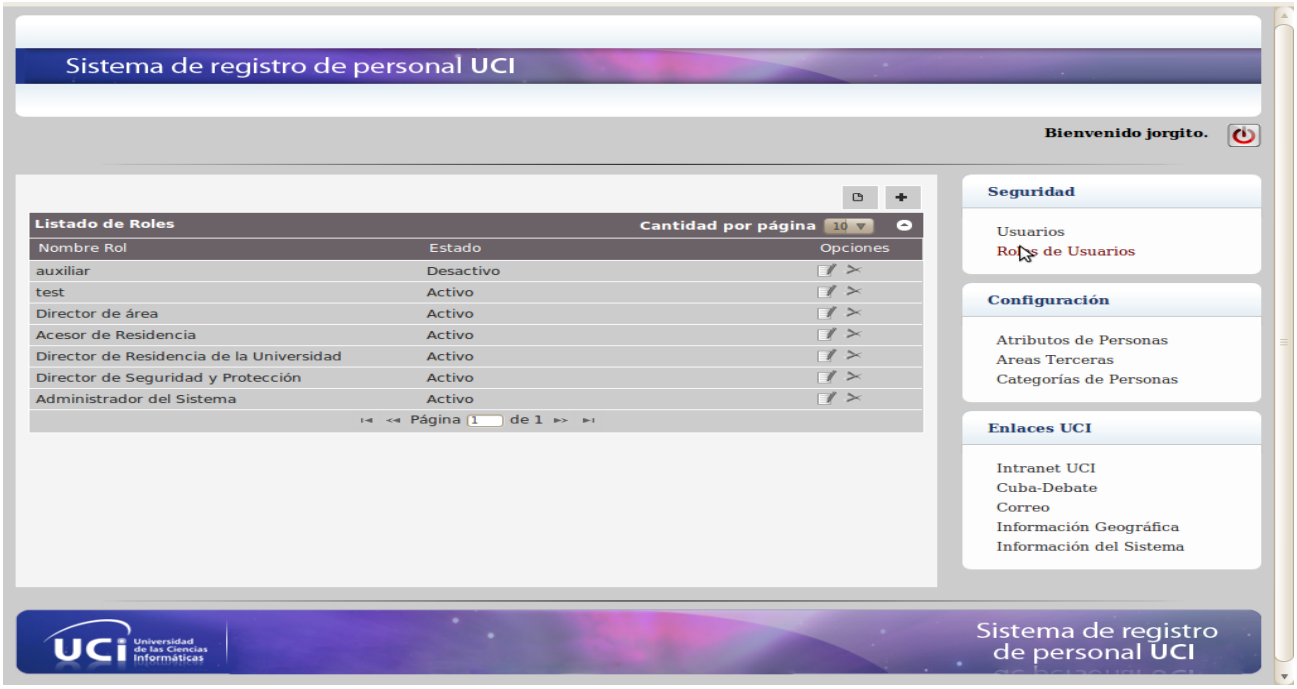
Anexo 10: Vista Modificar datos de persona



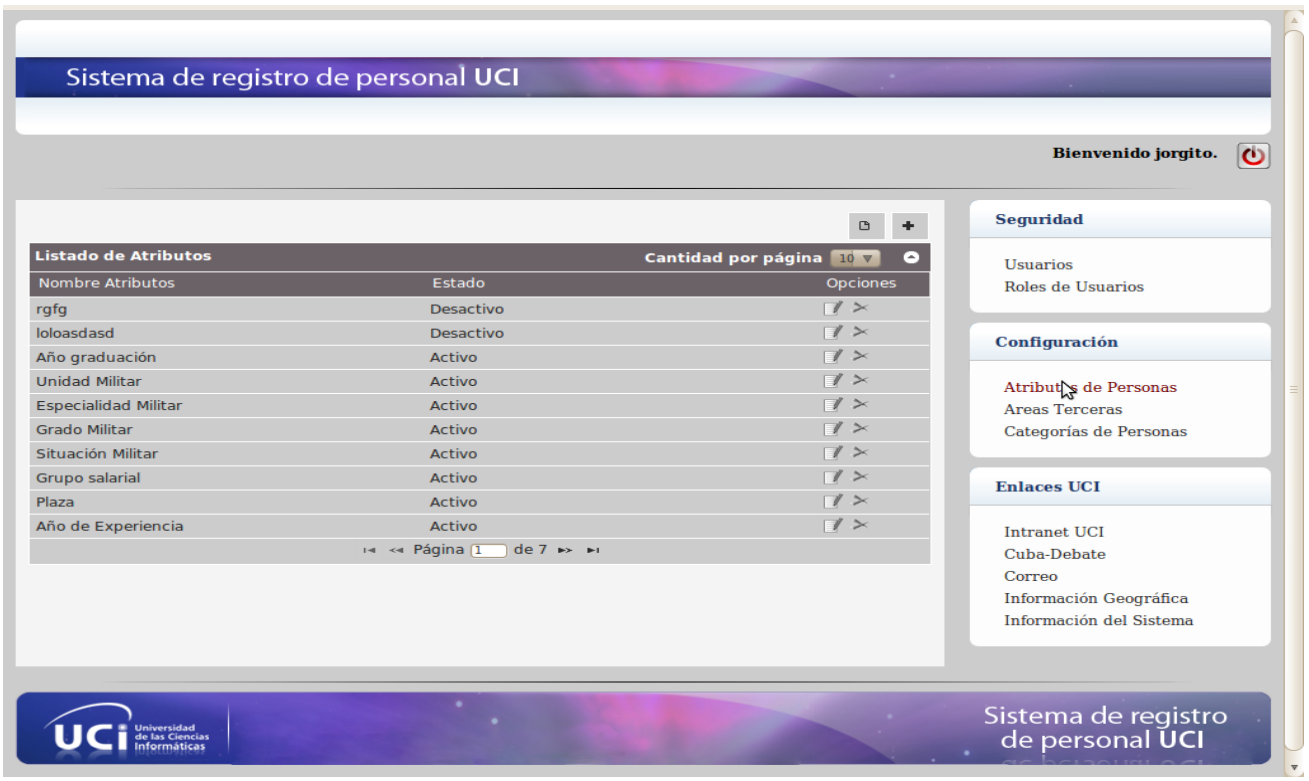
Anexo 11: Vista Dar baja



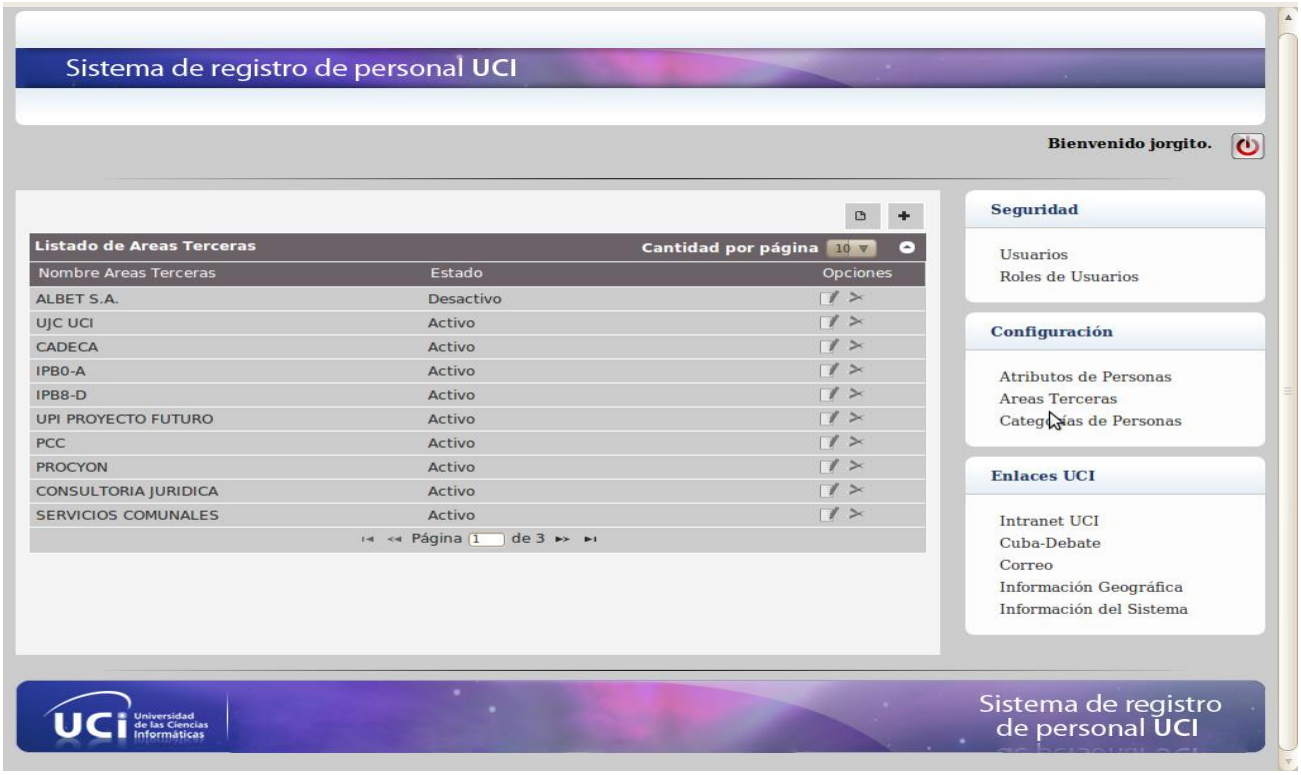
Anexo 12: Vista Gestionar usuario



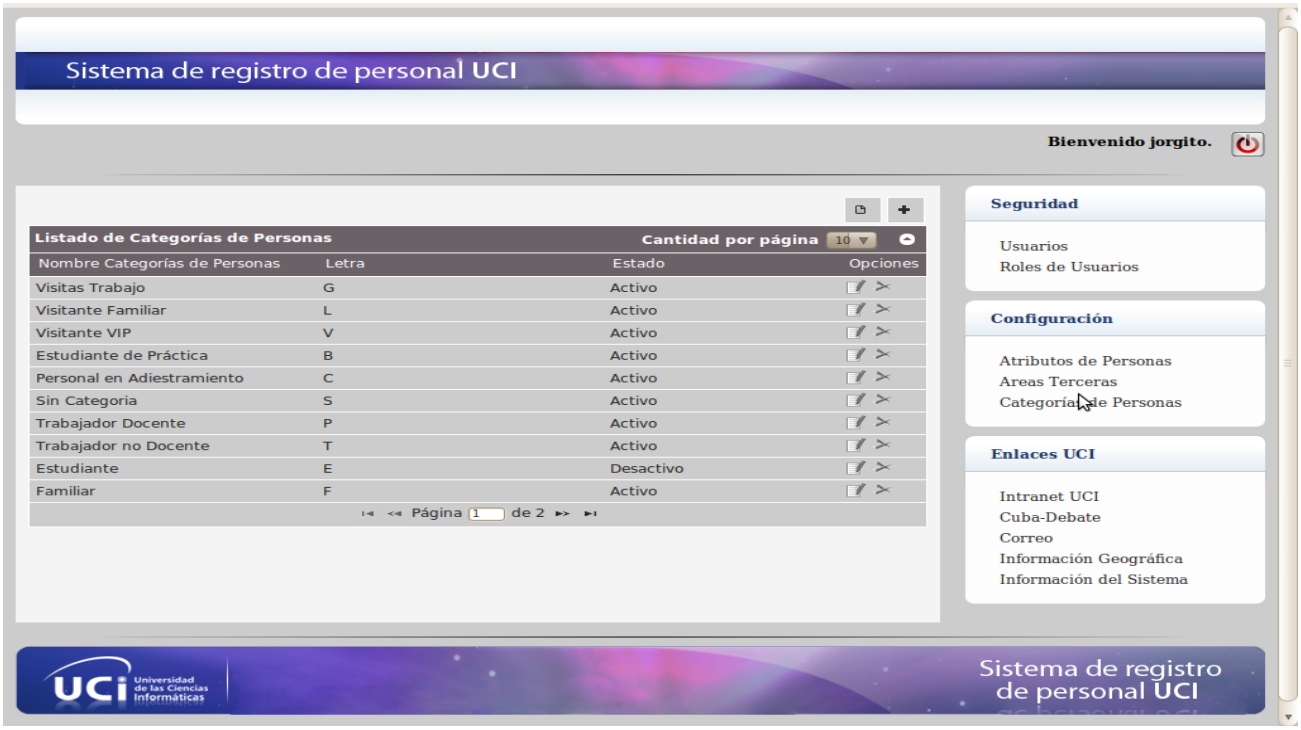
Anexo 13: Vista Gestionar nomenclador rol



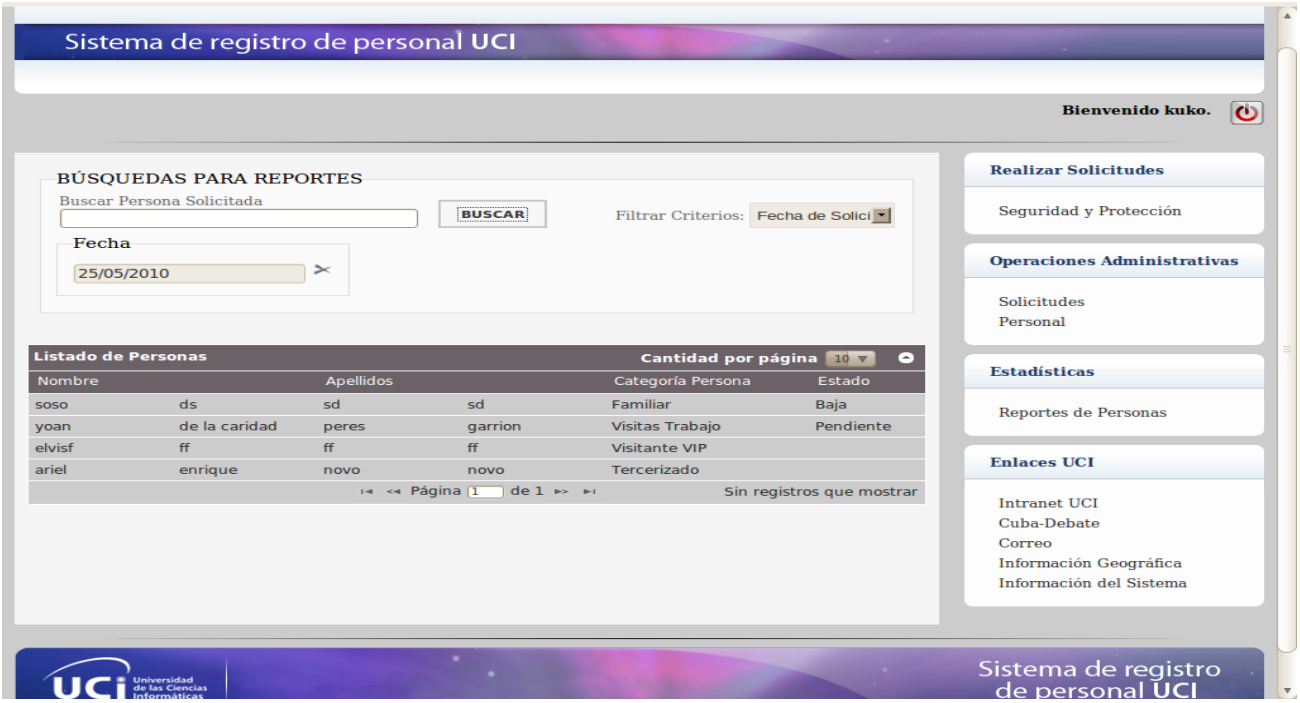
Anexo 14: Vista Gestionar atributos de persona



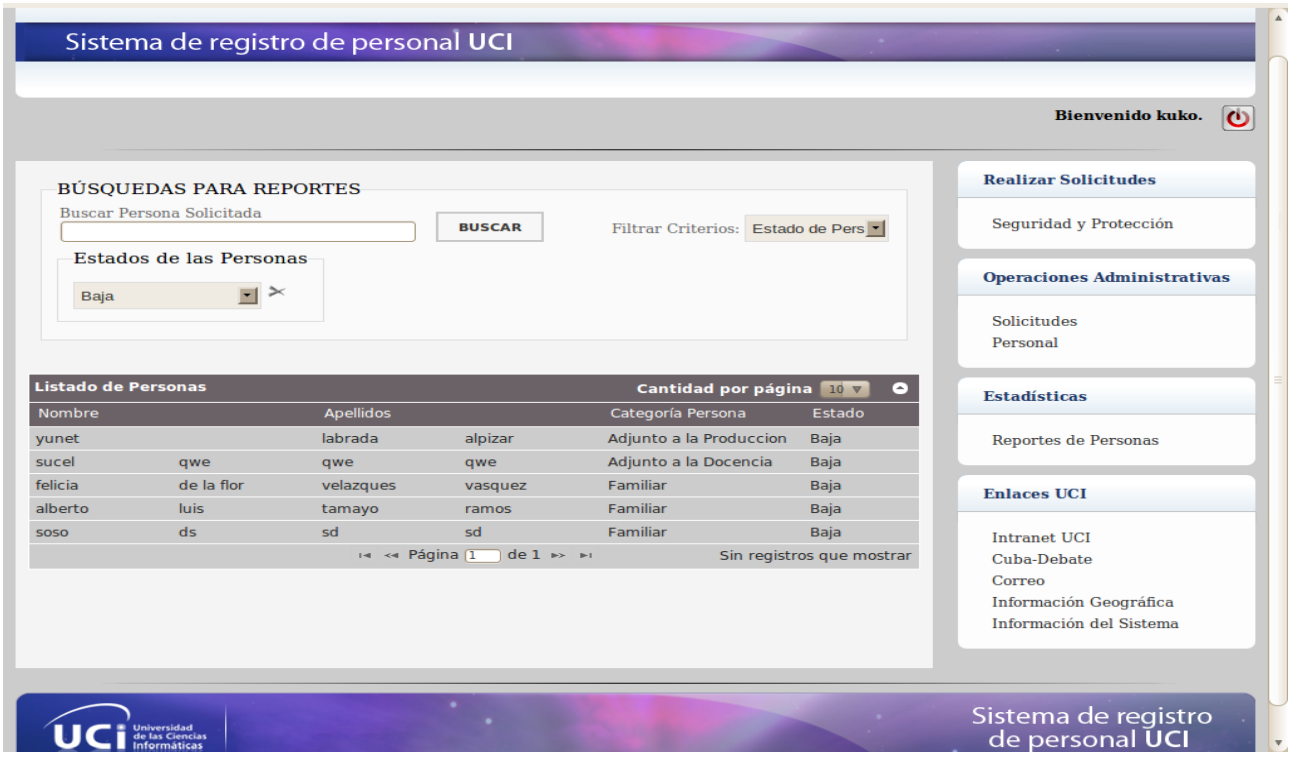
Anexo 15: Vista Gestionar nomenclador área tercera



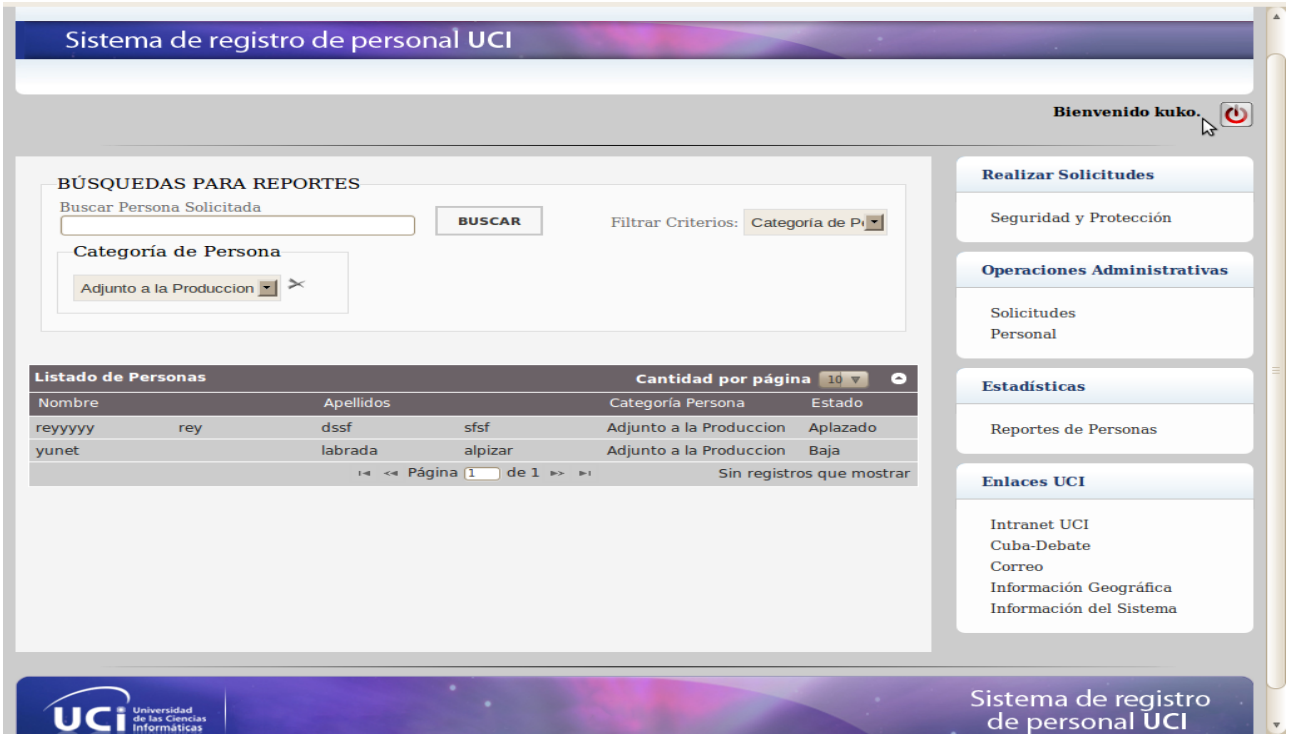
Anexo 16: Vista Gestionar nomenclador categoría de persona



Anexo 17: Vista Obtener solicitudes por fecha



Anexo 18: Vista Obtener personal por estado de persona



Anexo 19: Vista Obtener personal por categoría de persona

Historia de Usuario	
Nombre Historia de Usuario: *[Nombre que identifica la historia.]*	
Número: *[Número de la historia]*	**Referencia:** *[Se hace referencia a los requisitos que tiene en cuenta la HU]*
Programadores: *[Programador responsable de su implementación]*	**Iteración Asignada:** *[Que iteración se desarrollará. (Según su importancia)]*
Prioridad: *[Prioridad puede ser Alta, Media o Baja(Según Cliente)]*	**Puntos Estimados**: *[Tiempo en semanas que se le asignará. (Estimado)]*
Riesgo en Desarrollo: *[Riesgo puede ser Alto, Medio o Bajo (Según Programadores)]*	**Puntos Reales:** *[Tiempo real dedicado a la realización de la HU en semanas.]*
Descripción: *[Breve descripción del proceso que define la historia.]*	
Observaciones: *[Alguna acotación importante de señalar acerca de la historia.]*	

Prototipo de interfaz: *[Imagen de cada una de las interfaces relacionadas con la HU.]*

Anexo 20: Planilla de historia de usuario

	Tarea de ingeniería
Número de la tabla: *[Los números deben ser consecutivos]*	**Historias de Usuario implicadas:** *[Número de la historia de usuario a la que pertenece la tarea]*
Nombre de la tarea: *[Nombre que identifica a la tarea.]*	
Tipo de tarea: *[Las tareas pueden ser de: Desarrollo, Corrección, Mejora, Otra(Especificar)]*	**Puntos estimados:** *[Tiempo en semanas que se le asignará. (Estimado)]*
Fecha de inicio:	**Fecha de fin:**
Programador responsable: *[Nombre y Apellidos del programador]*	
Descripción: *[Breve descripción de la tarea.]*	

Anexo 21: Planilla tarea de ingeniería

GLOSARIO DE TÉRMINOS

Aplicación web: Aplicación que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador.

API: Aplication Programming Interface (en español *Interfaz de Programación de Aplicaciones*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Usados generalmente en las bibliotecas.

DOM: Es una abreviatura de Document Objet Model, es la estructura de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante JavaScript para cambiar dinámicamente los contenidos y aspecto de la página.

ECMAScript: Es un lenguaje de script definido por el estándar ECMA-262, la tecnología que dio origen a ECMAScript fue JavaScript.

Eventual: Trabajador temporal, no pertenece a la plantilla de una empresa y presta sus servicios de manera provisional.

Flowcharting: Un diagrama de flujo es un tipo común de diagrama, que representa un algoritmo o proceso, mostrando los pasos como cajas de varios tipos, y su orden mediante la conexión de éstos con las flechas. Esta representación esquemática puede dar una solución paso a paso para un problema dado. Los datos son representados en estos cuadros y flechas que conectan la representación flujo / dirección del flujo de datos. Diagramas de flujo se utilizan en el análisis, diseño, documentación o la gestión de un proceso o programa en varios campos.

Frameworks: Es una estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos o módulos de software concretos, sobre la base de un proyecto de software que puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Gestor de base datos: Es un tipo de software específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de manipulación de

datos y de un lenguaje de consulta. El propósito general de los gestores de bases de datos es manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos.

HTTP: Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web.

Identación: Significa mover un bloque de texto hacia la derecha insertando espacios o tabuladores para separarlo del texto adyacente, lo que en el ámbito de la imprenta se ha denominado siempre como sangrado o sangría.

JSON: Sigla de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Metodología ágil: Constituye una solución orientadas a software pequeños, aportando una elevada simplificación de la documentación para asegurar la calidad y el desarrollo rápido del producto.

Servidor web: Un servidor web no es más que un servidor que atiende pedidos siguiendo las especificaciones del protocolo HTTP. El servidor web recibe las peticiones de los clientes y pone en funcionamiento una serie de herramientas para darle respuesta a la solicitud. Debe comprobar que el recurso que se pide existe y si el usuario tiene autorización a accederlo. En dependencia de la plataforma en que corre el servidor web, se tendrá instalado un programa que lo representa y que se encarga de realizar todos estos procesos.

URL: Son las siglas de Localizador de Recurso Uniforme (en inglés *Uniform Resource Locator*), la dirección global de documentos y de otros recursos en la World Wide Web.

Web service: Un servicio web (en inglés, *Web* service) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

XML: Siglas de Extensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos.

XMLHttpRequest: También referida como XMLHTTP (Extensible Markup Language / Hypertext Transfer Protocol), es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web.