

Universidad de las Ciencias Informáticas

Facultad 2



Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Herramienta para la Gestión Empleadora del
Departamento de Capital Humano de la Universidad de las
Ciencias Informáticas.

Autores: Melquiades Megret Ramírez

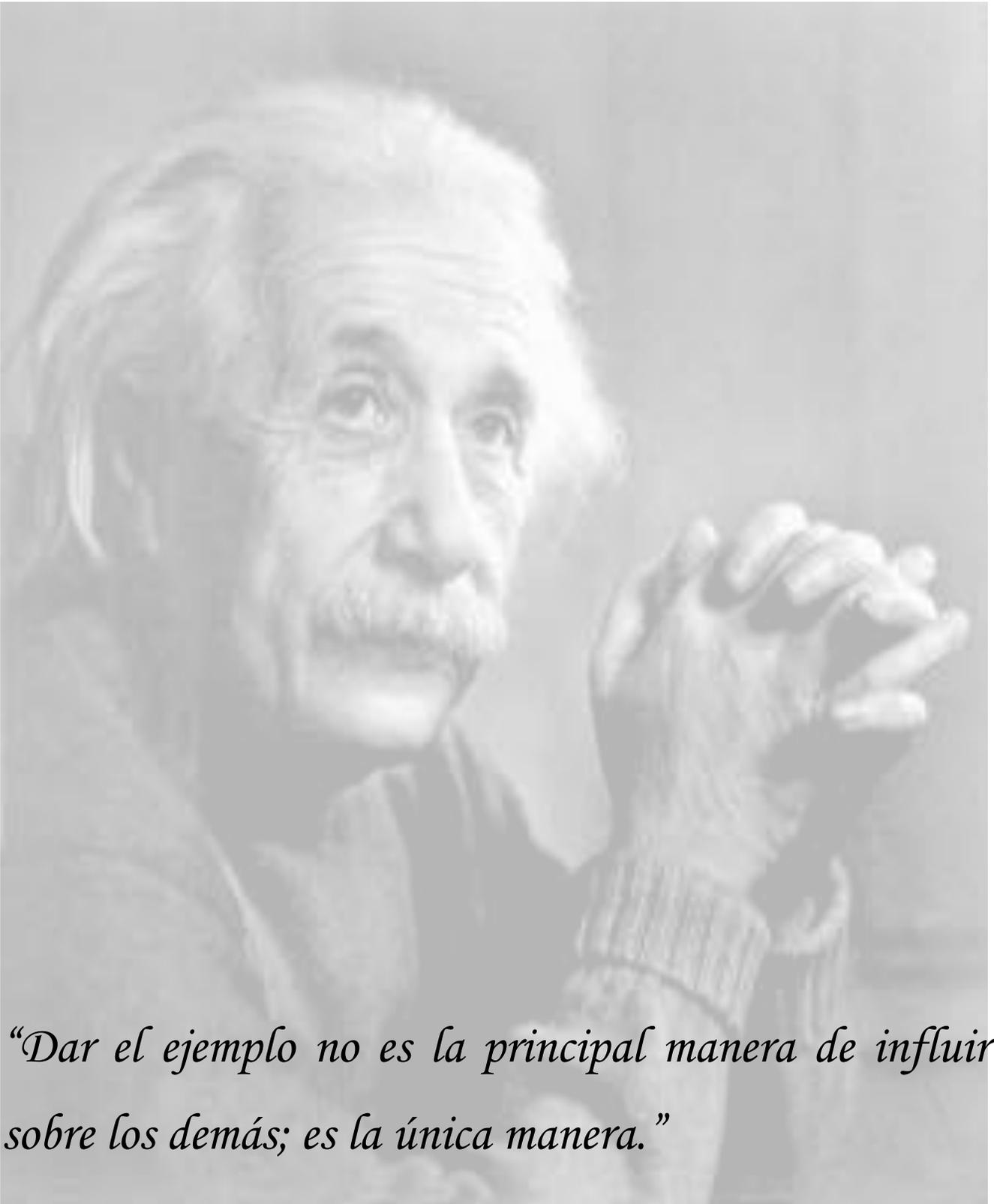
Jorge Ameth García Diago

Tutores: Ing. Husseyn Despaigne Reyes

Ing. Alberto Bradshaw González

La Habana, Junio del 2015

“Año 57 de la Revolución”



“Dar el ejemplo no es la principal manera de influir sobre los demás; es la única manera.”

Albert Einstein

DECLARACIÓN DE AUTORÍA



DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del presente trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Melquiades Megret Ramírez

Autor

Jorge Ameth García Diago

Autor

Ing. Husseyn Despaigne Reyes

Tutor

Ing. Alberto Bradshaw González

Tutor

DATOS DE CONTACTO

DATOS DE CONTACTO

Ing. Husseyn Despaigne Reyes (hdespaigne@uci.cu): Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2011. Se desempeña como especialista en el Centro de Telemática.

Ing. Alberto Bradshaw González (bradsg@uci.cu): Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2012. Se desempeña como especialista en el Centro de Telemática.

AGRADECIMIENTOS

Melquiades

Han sido varios años de estudio donde he tenido que dar lo mejor de mí. Siempre con el apoyo de mi seres queridos, amistades y la Revolución. Por eso siempre estaré agradecido con todos, porque más que mía ha sido una victoria suya.

Jorge Ameth

Agradecer además a mi Padre y a mi familia en general. A mis amigos. A todas aquellas personas que hicieron lo posible y lo imposible porque este momento tuviera lugar y también a todas aquellas que de una forma u otra intentaron impedirlo.

DEDICATORIA

Melquiades

Dedico esta tesis ante todo a mi mamá por ser mi guía.

A mi hermana que siempre ha sido mi modelo a seguir.

A mi padre por el apoyo incondicional.

A demás familiares por su preocupación e interés.

A mis amigos, que más que amigos han sido hermanos, los cuales siempre me han apoyados en todo momento.

Jorge Ameth

Eternamente, quisiera dedicarle este trabajo a mi madre. Sin la cual no sería nadie en esta vida y nada de lo que hasta hoy he logrado hubiera sido realmente un logro. Por haber estado a mi lado tanto tiempo y haber peleado tantas batallas por mí. A la vida y al Amor.



RESUMEN

El proceso de contratación que se realiza en los departamentos de capital humano de cualquier institución hoy en día es uno de los más importantes, ya que adquirir al personal mejor capacitado es lo que trae consigo el crecimiento, avance y desarrollo de cualquier empresa. En la Universidad de las Ciencias Informáticas mediante este proceso se maneja un gran volumen de información de forma manual y en papel, lo que implica que no cuente con la rapidez y efectividad necesaria.

La presente investigación tiene como objetivo desarrollar una herramienta informática que permita informatizar el proceso de contratación y seguimiento del capital humano en la Universidad. Para ello se utilizaron las herramientas de desarrollo PyCharm, el lenguaje de programación Python, el servidor HTTP Apache y como gestor de base de datos PostgreSQL. Por tal motivo se llevó a cabo un estudio de los sistemas informáticos existentes en el mundo, dirigidos al proceso de contratación. Además, se utilizó como metodología de desarrollo la Programación Extrema o eXtreme Programming (XP por sus siglas en inglés), que permitió un desarrollo iterativo e incremental, obteniéndose pequeñas mejoras en cada una de las etapas del proyecto.

Palabras Claves: contratación, empleo, recursos humanos, selección.



ÍNDICE

Tabla de contenido

DECLARACIÓN DE AUTORÍA	III
DATOS DE CONTACTO	IV
AGRADECIMIENTOS.....	V
DEDICATORIA.....	V
RESUMEN	VIII
ÍNDICE	VIII
Introducción.....	1
Capítulo 1. Fundamentación Teórica.....	6
1 Introducción.....	6
1.1 Marco Conceptual.....	6
1.1.1 Proceso de Contratación.....	6
1.1.2 La Selección de Capital Humano.....	8
1.1.3 La Gestión de Selección e Integración al Empleo del Capital Humano.....	8

ÍNDICE



1.1.4	La Gestión	9
1.1.5	Capital Humano	9
1.1.6	Proceso Productivo	9
1.2	Análisis de las soluciones existentes	9
1.2.1	Sistemas Internacionales	9
1.2.1.1	Mercadojobs	9
1.2.1.2	Computrabajo	10
1.2.1.3	Bumerán	11
1.2.1.4	Acciontrabajo	11
1.3	Técnicas, tecnologías y metodologías actuales en la que se apoya la solución del problema	12
1.3.1	Metodologías de Desarrollo	12
1.3.1.1	Metodología de Desarrollo de Software Seleccionada	14
1.3.2	Servidor Web	15
1.3.2.1	Servidor de Aplicaciones Seleccionado	16
1.3.3	Sistema Gestor de Base de Datos	17
1.3.3.1	Sistema Gestor de Base de Datos Seleccionado	17
1.3.4	Herramienta de Desarrollo	17

ÍNDICE



1.3.4.1	Herramienta de Desarrollo Seleccionada.....	18
1.3.5	Herramientas CASE.....	18
1.3.5.1	Visual Paradigm 8.0	19
1.3.6	Lenguaje de Programación.....	19
1.3.6.1	Lenguaje de Programación Seleccionado	19
1.3.7	Framework.....	19
1.4	Conclusiones	21
Capítulo 2 Características, Exploración y Propuesta del Sistema		22
2	Introducción.....	22
2.1.1	Proceso de contratación en el Departamento de Capital Humano de la Universidad de las Ciencias Informáticas.....	22
2.1.2	Herramienta para la Gestión Empleadora del Departamento de Capital Humano de la Universidad de las Ciencias Informáticas.....	24
2.2	Características no funcionales del sistema.....	25
2.2.1	Usabilidad.....	25
2.2.2	Disponibilidad.....	25
2.2.3	Hardware	25
2.2.4	Software.....	26

ÍNDICE



2.2.5	Interfaz de usuario.....	26
2.2.6	Soporte.....	26
2.3	Personas relacionadas con el sistema	26
2.4	Fase de Exploración.....	27
2.4.1	Historias de Usuarios (HU)	27
2.5	Planificación.....	33
2.5.1	Estimación de esfuerzo por Historias de Usuario.....	33
2.5.2	Plan de Iteraciones.....	34
2.5.2.1	Iteración 1	34
2.5.2.2	Iteración 2	34
2.5.2.3	Iteración 3	35
2.5.3	Plan de Duración de las Iteraciones	35
2.5.4	Plan de Entrega	36
2.6	Conclusiones	36
Capítulo 3. Diseño del Sistema		37
3	Introducción.....	37
3.1	Arquitectura.....	37

ÍNDICE

3.2	Patrones Arquitectónicos	38
3.2.1	Patrón Arquitectónico Modelo Vista Plantilla	38
3.3	Diagrama de Clases	40
3.3.1	Diagrama de Clases de la Aplicación Web	40
3.4	Patrones de Diseño	42
3.4.1	Patrones para Asignar Responsabilidades (GRASP).....	43
3.4.2	Patrones de diseño (GOF)	44
3.5	Tarjeta Clase-Responsabilidad-Colaborador.....	45
3.6	Diagrama de clases persistentes	46
3.7	Modelo Físico de Base de Datos.....	47
3.8	Conclusiones	48
Capítulo 4	Implementación y Prueba.....	49
4	Introducción.....	49
4.1	Estándares de Codificación.....	49
4.1.1	Formato de Código.....	49
4.2	Tareas de la Ingeniería	51
4.3	Pruebas	53

ÍNDICE



4.3.1 Pruebas Unitarias	53
4.3.2 Prueba de Aceptación	56
4.4 Conclusiones	59
Conclusiones Generales	60
Recomendaciones.....	61
Referencias Bibliográficas.....	63
Bibliografía	69
Anexos	75
Anexo 1 Historias de Usuarios	75
Anexo 2 Tarjeta Clase-Responsabilidad-Colaborador.....	Error! Bookmark not defined.
Anexo 3 Tareas.....	Error! Bookmark not defined.
Anexo 4 Pruebas de Aceptación	Error! Bookmark not defined.

Introducción.

La selección es un proceso que se ha llevado a cabo desde la comunidad primitiva, cuando el hombre tuvo la necesidad de agruparse para poder sobrevivir en su interacción con la naturaleza y transformarla en su beneficio. En este momento surge la selección con un carácter totalmente natural y espontáneo; a partir de ella se escogían las personas encargadas de realizar las diferentes actividades de la comunidad, teniendo en cuenta fundamentalmente la edad, el sexo y las características físicas (1).

El capital humano es un término usado en ciertas teorías económicas con el fin de determinar el crecimiento y designar a un hipotético factor de producción, dependiente, tanto de la calidad como la cantidad. Es factor determinante el grado de formación y de productividad de las personas involucradas en un proceso productivo. Este concepto ha sido extendido para designar el conjunto de recursos humanos que posee una empresa o institución económica. Igualmente se habla de modo informal de "mejora en el capital humano" cuando aumenta el grado de destreza, experiencia o formación de las personas de dicha institución económica (2).

La turbulencia y competitividad del marco empresarial actual exige de los sistemas de producción y de servicios, un capital humano mucho más preparado y comprometido, con la responsabilidad y creatividad suficientes para hacer suyo los problemas de la entidad y enfrentarlos con las soluciones que mejor respondan a cada situación en particular (3).

Una amenaza crítica en la gestión de recursos humanos es el aumento de los problemas en el personal, que representa un área de gran riesgo para el negocio. La pérdida de personal clave en tiempos difíciles, el alto costo de reclutamiento y entrenamiento de nuevos recursos humanos, todos pueden poner en peligro el negocio.

Las personas son el verdadero corazón de una organización. Una motivada y productiva fuerza de trabajo pueden hacer la diferencia en darle a una organización la ventaja por encima de sus competidores.

Adquirir trabajadores verdaderamente competentes constituye hoy una necesidad de primer orden para las organizaciones, porque son estos los principales protagonistas del ascenso de las empresas al éxito.

INTRODUCCIÓN

Proveer a las organizaciones del personal necesario para la realización de las actividades y al mismo tiempo encontrar alternativas para desarrollar y adecuar a los cargos el capital humano con que se cuenta, es uno de los principales retos que demanda la nueva gestión del capital humano (4).

La Universidad de las Ciencias Informáticas (UCI) cuenta con un Departamento de Capital Humano y con varias subdirecciones subordinadas al mismo. Este es el encargado de realizar diferentes funciones entre las que se encuentran, procesar las nóminas de trabajadores, gestionar las solicitudes de plazas vacantes existentes en las diferentes direcciones por departamentos, además de seleccionar y proponer a posibles candidatos de las plazas a ocupar.

Para comunicar a la comunidad universitaria que existen plazas vacantes, se publican anuncios en la Intranet; pero no todo el personal de la UCI accede a este portal y de esta forma la información no llega a un grupo considerable de personas. Las ofertas publicadas en la Intranet en ocasiones no llegan a ser vistas debido al abundante contenido que maneja este sitio. Además, como la promoción de ofertas de plazas no es el propósito principal de este portal, no se le da la prioridad requerida, siendo el aviso sustituido por otro de cualquier tipo, dígase noticias, convocatorias a eventos, entre otros, aun cuando continúa vigente la convocatoria.

También se pegan anuncios de papel en puertas y paredes de los diferentes locales o áreas de la Universidad como comedores, edificios docentes y cerca del departamento de Capital Humano; lo cual atenta contra la estética de los mismos. Además, estos anuncios no siempre son visibles y se deterioran debido a la manipulación de las personas o por factores climatológicos como el viento y la lluvia. Esto ocasiona que la información no llegue a un grupo considerable de personas y se pierda antes de que pueda cumplir con su objetivo principal que es el de informar.

Una de las vías para que el personal ajeno a la UCI se entere de la oferta de una plaza vacante, es el aviso de algún amigo o conocido, que sea trabajador de la Universidad. Limitando el número de personas que pueden enterarse de la oferta.

Una persona que desea optar por una plaza en la UCI, debe de dirigirse personalmente al Departamento de Capital Humano para informarse de las ofertas existentes y entregar la solicitud correspondiente en caso de estar interesado. En caso de no ser trabajador de la UCI, el aspirante debe viajar hasta la Universidad,

INTRODUCCIÓN

empleando tiempo y gastos en transporte, dependiendo del lugar donde radique. Además, para entrar a la UCI, previamente debe solicitar un permiso de entrada en la puerta de servicio, que en ocasiones demora en ser aprobada.

Si el interesado en una plaza vacante ya es trabajador de la UCI, también tendría que dirigirse personalmente a Capital Humano para entregar la solicitud. Empleando tiempo y esfuerzo para llegar hasta el local de este departamento. Las solicitudes son entregadas en papel y en ocasiones se pierden o sufren daños cuando son manipuladas, atentando contra su integridad.

En Capital Humano, el aspirante es atendido por un asistente, pero en ocasiones esta persona se encuentra realizando otras labores propias de su oficio. Esto puede traer consigo un retraso en la atención al aspirante y una acumulación de tareas pendientes para el asistente.

Durante el proceso de selección de la persona, los trabajadores de Capital Humano y del área que oferta la plaza deben comunicarse vía correo electrónico para notificar uno al otro, que ha concluido una etapa de este proceso. Esta forma de notificación, provoca un aumento en la cantidad de correos que reciben los encargados de la contratación; provocando demoras en dar respuesta a cada uno de estos correos.

Después de analizar la situación antes expuesta, se plantea como **Problema a Resolver** la siguiente interrogante: ¿Cómo garantizar una mayor efectividad en el proceso de contratación realizado por la Dirección de Capital Humano de la Universidad de las Ciencias Informáticas?

Se define como **Objeto de Estudio**, proceso de contratación de personal.

El **Campo de Acción** se centra en el proceso de contratación de personal en la Universidad de las Ciencias Informáticas.

Para dar solución a la problemática planteada se ha definido como **Objetivo General** de la investigación, desarrollar una herramienta informática que permita informatizar el proceso de contratación y seguimiento del Capital Humano en la Universidad de las Ciencias Informáticas.

Para dar cumplimiento al objetivo general se plantean las siguientes **Tareas de la Investigación**:

- ✓ Análisis de las regulaciones existentes para la contratación en la Universidad de las Ciencias Informáticas.

INTRODUCCIÓN

- ✓ Caracterización de los sistemas informáticos existentes a nivel nacional e internacional utilizados para la contratación de personal.
- ✓ Análisis de las técnicas y tecnologías para el proceso de contratación realizado en la Universidad de las Ciencias Informáticas.
- ✓ Modelado de la solución de software para la política de gestión de empleos y confección de solicitudes de empleos en la Dirección de Capital Humano ajustados a las características propias de la institución.
- ✓ Preparación y capacitación correspondiente para el uso de la arquitectura definida y aprobada por la Universidad para el desarrollo de las aplicaciones web.
- ✓ Implementación de una herramienta informática que pueda ser usada por la Dirección de Capital Humano de la UCI en el proceso de contratación.
- ✓ Realización de las pruebas a la aplicación funcional para detectar posibles errores y corregirlos.

Como base de apoyo para el desarrollo de la investigación se hace uso de los siguientes métodos científicos:

➤ **Métodos Teóricos:**

Análisis sintéticos: Este método permitirá analizar las teorías y los documentos referentes al objetivo de la investigación, facilitando de esta forma la extracción de los elementos más importantes relacionados con el objeto de estudio. Además de que posibilitará construir el camino a seguir, a partir del análisis detallado de cada uno de los documentos previamente mencionados.

Histórico-Lógico: Este método permitirá estudiar todo lo relacionado con el proceso de contratación que se realiza en los diferentes departamentos de capital humano, para así obtener un conocimiento histórico de su desarrollo y comportamiento tanto a nivel internacional como nacional.

Modelación: En el desarrollo se crearán reproducciones simplificadas de la realidad, por ejemplo, todos los modelos y diagramas presentados. Estos permitirán una reproducción ampliada de la realidad, además de que posibilitará descubrir y estudiar nuevas relaciones y cualidades del objeto de estudio.

➤ **Métodos Empíricos:**

Entrevista: Este método se utilizará para la realización del sistema, ya que para obtener el servicio con la calidad que requiere se realizarán una serie de entrevistas con el cliente, y sobre la base de estas se trabajará para satisfacer sus necesidades. De esta manera, se realizaron varias entrevistas con el cliente para definir los principales problemas existentes a los que se les daría solución posteriormente, identificando los procesos que intervienen en el proceso de contratación. Además, se entrevistó a varias personas especializadas en este proceso definiendo los problemas existentes a los que se les dará solución posteriormente.

Observación: Este método permitirá en aplicar los sentidos a un objeto o a un fenómeno, para estudiarlo tal como se presenta en realidad. Se utilizará en el diagnóstico del problema y luego en el diseño de la investigación. Con este método se logra reunir información visual sobre todo lo que ocurre respecto al problema que se intenta solucionar.

El presente trabajo de diploma consta de cuatro capítulos, donde se describe todo el proceso de la investigación

En el Capítulo I Fundamentación Teórica: Contiene el marco conceptual en el que se muestran las principales definiciones utilizadas en la investigación. Incluye un análisis de los sistemas que pudieran existir en la actualidad que se vinculan con la investigación y el estado del arte de las tecnologías, herramientas y metodologías que serán utilizadas en el desarrollo de la aplicación.

En el Capítulo II Características, Exploración y Propuesta del Sistema: se caracteriza el sistema, se reflejará la investigación realizada con los procesos que tienen lugar en el negocio. Se describirá la solución propuesta y se plantean a grandes rasgos las Historias de Usuarios (HU) que son de interés para la primera entrega del producto, así como una propuesta del prototipo funcional de la aplicación.

En el Capítulo III Diseño del Sistema: se define la arquitectura del sistema y con ello patrones arquitectónicos empleados. Se procederá a abordar las HU definidas y tareas de la ingeniería en cada iteración para luego hacer una revisión de las mismas.

En el Capítulo IV Implementación y Prueba: trata los aspectos relacionados con la construcción de la solución propuesta. Materializa la propuesta del sistema que da cumplimiento a los objetivos trazados con la investigación y la validación del mismo a través de las pruebas realizadas al sistema.

Capítulo 1. Fundamentación Teórica

1 Introducción

Este capítulo está dedicado a analizar conceptos que se deben conocer para tener un dominio básico de la situación planteada. Además de realizar un análisis detallado del estado del arte en cuanto a sistemas existentes, tecnologías, metodologías y herramientas utilizadas para el desarrollo del sistema propuesto.

1.1 Marco Conceptual

1.1.1 Proceso de Contratación

Un contrato es un acuerdo legal entre dos o más partes en el que una parte (el contratista) acuerda realizar ciertas tareas a cambio de una recompensa (o en términos legales, una consideración) pagada por una de las partes (los usuarios del servicio) (5). El proceso de contratación es, por lo tanto, una serie de interacciones entre los actores de un contrato de una Asociación Público – Privada (APP); esto a su vez lleva a la ejecución del contrato (entrega de los servicios requeridos). La negociación caracteriza todas las etapas del proceso de contratación (6).

Este proceso se caracteriza por:

- Integrar su expediente de trabajo cuando ya se aceptaron las partes.
- La contratación se llevará a cabo entre la organización y el trabajador.
- La duración del contrato será por tiempo indeterminado o determinado.
- El contrato deberá ser firmado por el director general, el responsable directo y el trabajador.

Fases del Proceso de Contratación

➤ Identificación de la Necesidad

El personal responsable de la contratación identifica una necesidad directamente o recibe la petición por parte de otros órganos de la administración, usuarios, sociedad, etc. En este momento puede considerar la

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

inclusión de criterios sociales y ambientales acordes con las prioridades de política social y ambiental de la administración.

➤ **Definición del Objeto de Contrato**

Se establece exactamente lo que se desea conseguir mediante el producto, servicio u obra contratada. El órgano contratante tiene plena libertad para definir lo que necesita.

➤ **Definición de las Especificaciones Técnicas**

Se fijan las características técnicas que los licitadores deben cumplir para que su oferta se ajuste a los requerimientos del objeto del contrato. Deben ser parámetros medibles que constituyan los criterios, para cuando una oferta no cumpla con ellos esta sea rechazada.

Para no constituir un obstáculo injustificado a la competencia y al comercio los criterios introducidos deben estar ligados al objeto del contrato; deben estar detallados en la documentación de la contratación y garantizar el acceso igualitario de todos los potenciales oferentes.

Si se hace referencia a estándares predefinidos, las ofertas que no posean el estándar pero puedan demostrar su equivalencia al mismo, deben incluirse en el procedimiento. Los licitadores pueden utilizar cualquier medio para probar su equivalencia con el estándar y son los poderes adjudicadores los que deben probar que dicha equivalencia no existe.

Se pueden introducir variantes cuando la forma de adjudicación sea la oferta económicamente más ventajosa. Las variantes deben estar expresamente autorizadas.

➤ **Selección de Suministradores – Definición de Criterios de Admisibilidad**

Se establecen las condiciones de capacidad económica, financiera y técnica que deben cumplir los licitadores para poder ser incluidos en el proceso de contratación. Sólo las ofertas de licitadores que cumplan con las condiciones de admisibilidad serán tenidas en cuenta en las siguientes fases del proceso.

➤ **Adjudicación del Contrato – Definición de Criterios de Adjudicación**

Después de evaluar cada oferta de forma individualizada, se tomarán en consideración sólo aquellas que cumplen las especificaciones técnicas. El órgano adjudicador comparará precios y evaluará la calidad de

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

los productos y servicios. Los criterios de adjudicación evalúan los niveles de calidad o desempeño del licitador en relación a las especificaciones de manera que se adjudicará el contrato al concurrente que cumpla con las especificaciones técnicas y haya ofrecido o bien el precio más bajo o bien la oferta económica más ventajosa desde el punto de vista de la autoridad contratante.

Los criterios de adjudicación deben ser relevantes para el objeto del contrato; respetar los principios de no discriminación, igualdad de trato y transparencia y estar publicados de forma clara en el anuncio del concurso.

➤ **Ejecución del Contrato – Definición de Condiciones de Ejecución**

La empresa que resulte adjudicataria deberá cumplir con unas condiciones previamente establecidas y anunciadas por el órgano contratante. Los poderes adjudicadores podrán exigir condiciones especiales en relación con la ejecución del contrato siempre que éstas sean compatibles con el derecho comunitario y se indiquen en el anuncio de licitación o en el pliego de condiciones. Las condiciones en que se ejecute un contrato podrán referirse, en especial, a consideraciones de tipo social y medioambienta (7).

1.1.2 La Selección de Capital Humano

La selección de capital humano es un proceso con carácter dirigido que tiene como objetivo captar, seleccionar e integrar al empleo a aquellos candidatos con mayores posibilidades de lograr altos niveles de desempeño; lo que está causalmente relacionado con su adecuada integración con las características presentes y futuras del entorno laboral, a través de las competencias laborales (8).

1.1.3 La Gestión de Selección e Integración al Empleo del Capital Humano.

El entorno cambiante que se acrecienta en los últimos años en las diferentes esferas, tecnológica, económica, política y social, ha conllevado a trabajar en el perfeccionamiento de la gestión empresarial, siendo necesario trazar estrategias con el propósito de crear y perfeccionar métodos y procedimientos que permitan alcanzar altos niveles de desempeño organizacional. Se ha evidenciado que en este desempeño el factor humano juega un papel fundamental ya que es el que le confiere la ventaja competitiva a las organizaciones, razón por la cual cobra auge en la actualidad el proceso de selección e integración al empleo (9).

1.1.4 La Gestión

La gestión no es más que planificar, organizar y controlar. Es dirigir las acciones que constituyan la puesta en marcha concreta de la política general de la empresa, es tomar decisiones orientadas a alcanzar los objetivos marcados. La gestión es un proceso dinámico, interactivo, eficiente y eficaz; consistente en planear, organizar, liderar y controlar las acciones en la entidad, desarrollado por un órgano de dirección que cuenta con grupos de personas, recursos y autoridad para el establecimiento, logro y mejora de los propósitos de constitución de la organización, sobre la base del conocimiento de las leyes y principios, de la sociedad, la naturaleza humana y la técnica, así como de información en general (10).

1.1.5 Capital Humano

El capital humano es un término usado en ciertas teorías económicas del crecimiento para designar una mano de obra calificada y capacitada como factor de producción dependiente no solo de la cantidad, sino también de la cantidad, del grado de formación y de la productividad de las personas. Es el aumento en la capacidad de la producción del trabajo, alcanzada con mejoras en las capacidades de trabajadores. (11)

1.1.6 Proceso Productivo

Es un sistema de acciones que se encuentran interrelacionadas de forma dinámica y que se orientan a la transformación de ciertos elementos. De esta manera, los elementos de entrada (conocidos como factores) pasan a ser elementos de salida (productos), tras un proceso en el que se incrementa su valor (12).

1.2 Análisis de las soluciones existentes

1.2.1 Sistemas Internacionales

Actualmente en el mundo existen diversas herramientas para el proceso de contratación que se realizan en las diferentes empresas. Se conocen sistemas informáticos que están a disposición de cualquier empresa y cualquier usuario, con el objetivo de que dicho proceso de contratación proporcione una mayor facilidad tanto a las empresas, como a los empleados. Cada uno de ellos enfocados a las particularidades de la institución en la cual ha sido implementada.

1.2.1.1 Mercadojobs

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

Mercadojobs (www.mercadojobs.com) es un sitio que reúne vacantes de las mejores bolsas de empleo en el mundo, para facilitar la búsqueda a los usuarios de Internet. Está presente en más de 65 países superando más de 100,000 visitas diarias y con base a de datos de más de 6 millones de usuarios a quienes se les envían ofertas de empleo por correos diariamente. Este sitio es propiedad de BECITHEW SA DE CV, una empresa privada dentro de servicios de programación informática personalizada en Puebla. Ofrece servicios de búsqueda de empleos, alerta de empleos y publicidad donde los datos personales tratados son los ofrecidos voluntariamente por los usuarios para el uso de los servicios ofrecidos por el sitio. No se conoce la presencia en el sitio de contenidos ilícitos o no autorizados y, de todos modos, se compromete a quitar inmediatamente del sitio eventuales contenidos que fuesen lesivos a derechos de terceros, tras indicación escrita por parte del interesado. Mercadojobs se jacta de una accesibilidad alta del sitio y se esfuerza continuamente por mejorar el nivel de los propios servicios. En caso de interrupción temporal o prolongada de los servicios, Mercadojobs efectuará todo lo que esté a su alcance para limitar el tiempo de interrupción de los servicios, sin que ello pueda garantizar el tiempo necesario para restablecer los servicios (13).

1.2.1.2 Computrabajo

Computrabajo (www.computrabajo.com) es la red de web de trabajo líder en América Latina. Es la web de trabajo más visitada en Colombia, Perú y Costa Rica, y la segunda de México, Chile y Argentina. Además está presente en Bolivia, Cuba, República Dominicana, Ecuador, El Salvador, Guatemala, Honduras, Nicaragua, Panamá, Paraguay, Puerto Rico, Uruguay y Venezuela. Más de 36 millones de visitas mensuales. Actualmente, la red de computrabajo.com cuenta con 280 millones de páginas vistas y más de 36 millones de visitas mensuales. Fundado en 1999, computrabajo.com fue creada hace 15 años en Edimburgo, Escocia, donde ha tenido su sede durante muchos años. Esta web propietaria ofrece a las empresas herramientas avanzadas para la gestión de los procesos de selección y facilita a los profesionales el acceso a nuevas oportunidades de trabajo. Computrabajo actúa como conector entre personas y empresas. A través del sitio, las empresas buscan candidatos para sus puestos de trabajo, y los candidatos se postulan a las ofertas enviando su currículum. Para tener acceso a las ofertas que se publican en Computrabajo sólo es necesario que el usuario se registre e ingresar su currículum a través del formulario que se pondrá a tu disposición. Además la misma no interfiere en ningún momento en el proceso de selección de los candidatos (14).

1.2.1.3 Bumerán

Bumerán (www.bumeran.com) se especializa en el desarrollo de soluciones tecnológicas para los procesos de recursos humanos de los clientes. Reconocidos en el mercado como líderes en reclutamiento a través de Internet y en el desarrollo de tecnología para automatizar procesos internos relativos a la gestión de los recursos humanos. Toda la información permanece actualizada las 24 horas del día los 365 días del año a total disposición de los clientes según lo necesiten. A través del portal de Bumerán se les facilita a los usuarios el acceso y la utilización de diversos servicios y contenidos relacionados con la búsqueda de empleo. Se insertan en el portal ofertas de empleo suministradas por terceras empresas que deseen iniciar un proceso de selección, gestionando una base de datos de personas demandantes de empleo. Estas empresas que voluntariamente deseen formar parte de la misma y suministren sus datos a este fin, y publicando contenidos sobre cuestiones vinculadas al mundo laboral. Los datos son puestos a disposición de los usuarios por Bumeran, por terceros usuarios del portal y/o por terceros proveedores de servicios y contenidos, así como cualquier otro producto, servicio o contenido accesible a través del portal (15).

1.2.1.4 Acciontrabajo

Este web pertenece a la compañía W3 Ltd., registrada oficialmente en el Reino Unido desde el año 2001, cumple con las leyes del Reino Unido exclusivamente. Los datos aportados tanto por empresas clientes como candidatos a puestos de trabajo son gestionados por W3 Ltd. con la finalidad de ofrecer un enlace entre empleadores y trabajadores. Las empresas que hacen uso de este sitio web acuerdan al abrir una cuenta de empresa, el uso de los servicios ofrecidos con la única finalidad de ser utilizados como instrumento de ayuda en los procesos de selección abiertos para sí mismas o para terceros que hayan depositado en estas sus labores de selección de personal. Las empresas se comprometen también a ofrecer amplia información sobre las ofertas de empleo anunciadas. Asimismo, cualquier oferta de empleo que contenga datos que potencien el contacto directo del candidato con la empresa dejarán de ser publicadas y la práctica reiterada de esta conducta cualificará a W3 Ltd. a dar de baja a la empresa de todos los servicios ofrecidos por este web. Las empresas y candidatos que usan este sitio web podrán en cualquier momento actualizar y/o eliminar la totalidad de sus datos de la base de datos de este sitio a través de la web. En el caso de observar anomalías en el funcionamiento del sistema el candidato o la empresa puede solicitar a

través del contacto indicado al final de esta página la eliminación o actualización de sus datos de forma manual (16).

Conclusiones del estudio de las soluciones existentes.

Después de haber analizado las soluciones informáticas existentes para lo gestión y control de las diferentes operaciones que se llevan a cabo en el proceso de contratación en los Departamentos de Capital Humano, se puede concluir que no existe ninguna herramienta a nivel nacional para gestionar el proceso de contratación de personal. Mientras que los sistemas encontrados a nivel internacional, a pesar de que brindan una amplia gama de funcionalidades, no son factibles para su utilización en la Universidad de las Ciencias Informáticas debido a los siguientes argumentos:

- En su mayoría, son propietarios con altos costos por adquirirlos, implementarlos y personalizarlos.
- Han sido diseñados para empresas capitalistas con un modelo de gestión muy diferente a la mayoría de las empresas y entidades presupuestadas cubanas, donde la economía es centralizada y se requieren además algunos mecanismos de control de los recursos.
- Los sistemas no cubren las necesidades del personal referente al proceso de contratación debido a que las funcionalidades que brindan no están acorde al flujo actual de datos que se procesan en los diferentes departamentos subordinados al de Capital Humano. Además, muchos han sido desarrollados para una institución en específico por lo que no pueden extenderse hacia todos los servicios de que efectúa el proceso de contratación de la Universidad.

1.3 Técnicas, tecnologías y metodologías actuales en la que se apoya la solución del problema.

1.3.1 Metodologías de Desarrollo

Las metodologías de desarrollo son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevos software (21). Existen varios tipos de metodologías como las tradicionales y las ágiles. Entre las tradicionales se encuentra el Proceso Unificado de Desarrollo (RUP por sus siglas en inglés), que se centra especialmente en el control del proceso,

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir y las herramientas y notaciones que se usarán.

Entre las ágiles se encuentran Programación Extrema (XP por sus siglas en inglés), Scrum, Crystal, Evolutionary Project Management (Evo), Feature Driven Development (FDD) y Adaptive Software Development (ASD). Las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Actualmente se dice que están revolucionando la manera de producir software (22).

En la actualidad es de suma importancia seleccionar una metodología de desarrollo de software que esté acorde con las necesidades que requiera un proyecto. Pues es necesario tener en cuenta las propiedades de la metodología a seleccionar, para tener una visión clara de su funcionamiento respecto a los objetivos del proyecto. Para llevar a cabo esta selección se debe tener en cuenta una serie de parámetros como son las características del equipo de desarrollo y características del cliente.

El equipo de desarrollo tuvo en cuenta las siguientes características para seleccionar la metodología de desarrollo.

Tabla 1 Características del equipo de desarrollo.

Características	Valoración
Composición del equipo de desarrollo	2 personas
Experiencia productiva	(baja, media, alta) <ul style="list-style-type: none">• Media
Conocimiento de las metodologías ágiles	(bajo, medio, avanzado) <ul style="list-style-type: none">• Medio

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA



Conocimiento de las técnicas de programación	(bajo, medio, avanzado) <ul style="list-style-type: none">• Medio
--	---

Tabla 2 Características del cliente.

Características	Valoración
Cliente	Se encuentra dentro de la universidad interactuando de forma frecuente con el equipo.
Relación con el cliente	No forma parte del equipo de desarrollo.
Disponibilidad de tiempo	8 horas al día, de lunes a viernes.
Retroalimentación cliente-equipo desarrollo	(continua, discontinua) <ul style="list-style-type: none">• Continua

Se decide utilizar la metodología ágil XP para el desarrollo del presente trabajo ya que se adapta al desarrollo del proyecto y a las condiciones en que se realiza el mismo.

1.3.1.1 Metodología de Desarrollo de Software Seleccionada

XP

XP es una metodología de desarrollo de la ingeniería de software. Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un

aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Se puede considerar la Programación Extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software. Estos son los valores originales de la programación extrema: simplicidad, comunicación, retroalimentación coraje y respeto (23).

Características fundamentales (24)

- Desarrollo iterativo e incremental. Pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. La mayor calidad del código escrito de esta manera. El código es revisado y discutido mientras se escribe. Es más importante que la posible pérdida de productividad inmediata.
- Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo (25).

1.3.2 Servidor Web

En informática, se denomina servidor de aplicaciones a un servidor en una red de computadores que ejecuta ciertas aplicaciones. Usualmente se trata de un dispositivo de software que proporciona servicios de aplicación a las computadoras clientes. Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones. Los servidores de aplicación típicamente incluyen también middleware (o software de conectividad) que les permite intercomunicarse con variados servicios, para efectos de confiabilidad, seguridad, no-repudio, etc. Los servidores de aplicación también brindan a los desarrolladores una Interfaz para Programación de Aplicaciones (API por sus siglas en inglés), de tal manera que no tengan que preocuparse por el sistema operativo o por la gran cantidad de interfaces requeridas en una aplicación web moderna (26).

1.3.2.1 Servidor de Aplicaciones Seleccionado

HTTP Apache

El Servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. La arquitectura del servidor Apache es muy modular. El servidor consta de una sección núcleo y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web. Apache es usado primariamente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web (27).

Apache es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y lenguajes de programación PHP/Perl/Python (y ahora también Ruby). Este servidor web es redistribuido como parte de varios paquetes propietarios de software, incluyendo la base de datos Oracle y el IBM WebSphere application server. Mac OS X integra apache como parte de su propio servidor web y como soporte de su servidor de aplicaciones WebObjects. Es soportado de alguna manera por Borland en las

herramientas de desarrollo Kylix y Delphi. Apache es incluido con Novell NetWare 6.5, donde es el servidor web por defecto, y en muchas distribuciones Linux (27).

1.3.3 Sistema Gestor de Base de Datos

Hoy en día las empresas e instituciones están en constante procesamiento de información, la cual, para una mejor manipulación y acceso necesita de ser almacenadas en las diferentes formas de almacenamiento, ya sea en formato duro o digital, pero no cabe duda que una de las más seguras y conocidas son las bases de datos. Una base de datos es un almacén que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente. Una base de datos es el conjunto de datos informativos organizados en un mismo contexto para su uso y vinculación (28).

1.3.3.1 Sistema Gestor de Base de Datos Seleccionado

PostgreSQL 8.3

Es un potente servidor de base de datos objeto relacional de código abierto. Iniciado como un proyecto universitario en la década de 1980, ha llevado varias décadas de desarrollo y cuenta con una arquitectura confiable, estabilidad de procesamiento e integridad en el manejo de datos. Cuenta con una gran documentación, siendo este altamente escalable, tanto en la enorme cantidad de datos que puede manejar como en el número de usuarios concurrentes que puede acomodar. En PostgreSQL aquellos administradores de bases de datos que han tenido experiencia con otros manejadores de bases de datos podrán adaptarse rápidamente a su uso. Incorpora características modernas (orientado a objetos, manejo de bloqueos predefinido). Es altamente configurable, con lo cual puede personalizarse de acuerdo al escenario donde será utilizado. Es multiplataforma (cuenta con versiones para sistemas operativos basados en Unix y Windows). Es un verdadero proyecto de software abierto, ya que su desarrollo no es dirigido por una empresa sino es administrado por una comunidad (29).

1.3.4 Herramienta de Desarrollo

Las Herramientas de Ayuda al Desarrollo de Sistemas de Información, surgieron para intentar dar solución a los problemas inherentes a los proyectos de generación de aplicaciones informáticas: plazos y presupuestos incumplidos, insatisfacción del usuario, escasa productividad y baja calidad de los desarrollos,

entre otros. Algunas de estas herramientas se dirigen principalmente a mejorar la calidad. Se puede definir a las herramientas de desarrollo como un conjunto de métodos, utilidades, técnicas, programas informáticos, que usa un programador, para crear, depurar, gestionar o dar asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software (30).

1.3.4.1 Herramienta de Desarrollo Seleccionada

PyCharm 4.0.4

Como todos las demás IDE de JetBrains PyCharm presenta un editor de códigos inteligente, que entiende los detalles específicos de Python y ofreciendo extraordinarios mejoradores de la productividad como son formateo automático de código, finalización de código, refactorizaciones, importación automática y navegación de código con un solo clic, entre otros. PyCharm es multiplataforma, hay binarios para Windows, Linux y Mac OS X. Existen dos versiones de PyCharm una comunitaria y otra profesional con diferencias en características más que todo de en qué se integra una versión y otra (31).

Ventajas

- Autocompletado, resaltador de sintaxis, herramientas de análisis y refactorización.
- Integración con frameworks web como Django, Flask, Pyramid, Web2Py.
- Depurador avanzado de Python y JavaScript.
- Integración con lenguajes de plantillas Mako, Jinja2, Django Template.
- Soporte de entornos virtuales e intérpretes de Python 2.x, 3.x, PyPy, Iron Python y Jython.
- Compatibilidad con SQLAlchemy (ORM), Google App Engine, y Cython.
- Soporte para modo VIM con plugin.
- Soporta sistemas de control de versiones: Git, CVS, Mercurial.

1.3.5 Herramientas CASE

Las herramientas Ingeniería de Software Asistida por Computadora (CASE por sus siglas en inglés), son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el costo de las misma en términos de tiempo y costo. Estas herramientas pueden ayudar en

todos los aspectos del ciclo de vida del desarrollo de software, en tareas como el proceso de realizar un diseño de un proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detención de errores entre otras (32).

1.3.5.1 Visual Paradigm 8.0

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación (33).

1.3.6 Lenguaje de Programación

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana (34).

1.3.6.1 Lenguaje de Programación Seleccionado

Python 3.0

Es un lenguaje de programación de alto nivel que se desarrolla actualmente como un proyecto de código abierto, administrado por la Python Software Foundation. Entre los objetivos que busca este lenguaje se encuentra la facilidad, tanto de lectura, diseño, como de extensión. Posee soporte e integración con otros lenguajes y herramientas, por lo que se pueden escribir nuevos módulos fácilmente en C o C++, además tiene integradas varias bibliotecas estándar. Posee gran sencillez y velocidad, soporta diversas bases de datos y sus bibliotecas disminuyen gran parte del trabajo. Sin embargo, es preciso mencionar que al generar programas interpretados, estos son más lentos que los compilados (35).

1.3.7 Framework

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. También se encuentran frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, entre otros. En general, con el término framework, se hace referencia a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación.

En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le pueden añadir las últimas piezas para construir una aplicación concreta. Los mismos persiguen los siguientes objetivos: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un framework web, por tanto, se puede definir como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web (36).

1.3.7.1 Framework Seleccionado

Django 1.8

Django es un framework de desarrollo web de código abierto, escrito en Python, que cumple en cierta medida el paradigma del Modelo Vista Controlador. Fue desarrollado en origen para gestionar varias páginas orientadas a noticias, la meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, así como el desarrollo rápido de las aplicaciones (37).

Características

- ✓ Posee mapeado objeto relacional.
- ✓ Contiene una interfaz de programación de aplicaciones (API por sus siglas en inglés) de base de datos robusta.
- ✓ Incorpora un sistema de vistas genéricas que ahorra tener que escribir la lógica de ciertas tareas comunes.
- ✓ Posee un sistema extensible de plantillas basado en etiquetas, con herencias de plantillas.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

- ✓ Posee un soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.
- ✓ Presenta una documentación accesible a través de la aplicación administrativa.

En Django la base de datos más recomendada es PostgreSQL, pero también son soportadas MySQL y SQLite. Se encuentra en desarrollo un adaptador para Microsoft SQL Server. Una vez creados los modelos de datos, Django proporciona una abstracción de la base de datos a través de su API que permite crear, recuperar, actualizar y borrar objetos. También es posible que el usuario ejecute sus propias consultas SQL directamente. En el modelo de datos de Django, una clase representa un registro de una tabla en la base de datos y las instancias de esta serán las filas en la tabla. Incluye un servidor web liviano para realizar pruebas y trabajar en la etapa de desarrollo. En la etapa de producción, sin embargo, se recomienda Apache 2 con mod_Python, aunque Django soporta la especificación Web Server Gateway Interfaz (WSGI por sus siglas en inglés), por lo que puede correr sobre una gran variedad de servidores como FastCGI o SCGI en Apache u otros servidores (38).

1.4 Conclusiones

En este capítulo, se analizaron los principales conceptos relacionados con la gestión empleadora, los cuales son necesarios para la comprensión de la presente investigación. Se realizó el análisis de los sistemas existentes a nivel nacional e internacional relacionados con el campo de acción, demostrando que ninguno representa una solución a la problemática planteada, dado que no se ajustan a las necesidades de la Universidad de las Ciencias Informáticas. También se seleccionó la metodología, el sistema gestor de bases de datos, la herramientas de desarrollo y el lenguaje de programación necesarios para desarrollar la aplicación teniendo en cuenta las características de la misma.

Capítulo 2 Características, Exploración y Propuesta del Sistema

2 Introducción

En este capítulo se describen de manera general las características del sistema a desarrollar. Se identifican los procesos existentes con el objetivo de comprender aquellos que intervienen en la gestión empleadora del sistema de Capital Humano, para un correcto diseño e implementación de los mismos. También se hace alusión a las fases de Exploración y Planificación propias de la metodología de desarrollo utilizada, donde se confeccionan las HU importantes para cada iteración definida por el equipo de desarrollo.

2.1.1 Proceso de contratación en el Departamento de Capital Humano de la Universidad de las Ciencias Informáticas.

El proceso comienza cuando la dirección de un área o departamento necesita ocupar una plaza vacante. Esta debe ser presentada a la Dirección de Capital Humano con los fundamentos necesarios que validen la necesidad de ocupar la nueva plaza. Capital Humano lanza la convocatoria de las plazas aprobadas, en puntos de mucha fluctuación de trabajadores de la Universidad. La persona interesada en algunas de las ofertas publicadas se dirige al Departamento de Capital Humano, donde entrega una serie de datos, creando de esta manera una solicitud para ocupar dicha plaza.

Las solicitudes son enviadas a los distintos departamentos, los cuales se encargan de verificar los requisitos en el expediente laboral. Una vez determinadas las personas que reúnen los requisitos para ocupar la plaza vacante, se selecciona al más idóneo según sus competencias laborales. Posteriormente se notifica a la persona de que fue seleccionado, para que reúna los documentos pertinentes y los presente en el Departamento de Capital Humano para ser contratado.

A continuación se muestra el modelo del siguiente proceso:

CAPÍTULO 2 CARACTERÍSTICAS, EXPLORACIÓN Y PROPUESTA DEL SISTEMA

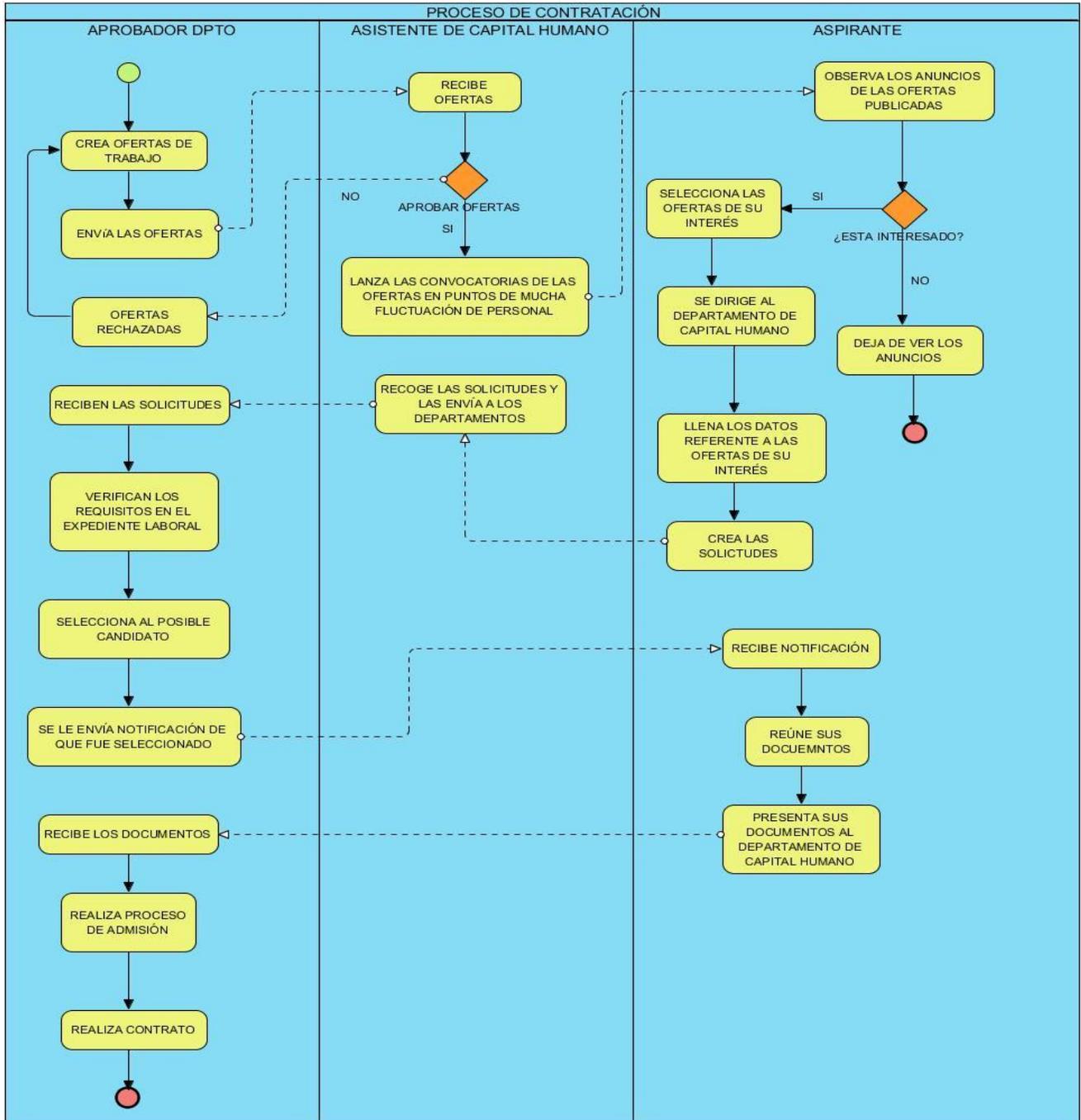


Figura 2 Proceso de Contratación

2.1.2 Herramienta para la Gestión Empleadora del Departamento de Capital Humano de la Universidad de las Ciencias Informáticas.

El proceso de contratación se inicia, cuando surge la necesidad de que una plaza determinada sea ocupada en uno de los departamentos o áreas de la Universidad. Estos proponen las plazas vacantes y crean las ofertas, aunque no serán publicadas en el sitio hasta que el departamento de Capital Humano las apruebe. Una vez aprobadas estas serán publicadas. Las ofertas también pueden ser rechazadas y en tal caso, quedarán pendientes de revisión o no se publicarán.

Una vez aprobadas las ofertas, se publican en el sitio para que estas sean vistas por los usuarios interesados. Los usuarios que acceden al sistema en busca de ofertas, deben llenar su currículum especificando algunos datos personales y generales, ejemplo: (ci, sexo, estado civil, organizaciones, nombre de los padres), currículum (desvinculado, recién graduado, jubilado, jubilado contratado), categorías (obrero, técnico, licenciado, ingeniero, master, doctor), entre otros. Luego de especificar dichos datos los aspirantes están en la capacidad de realizar solicitudes.

Cada aprobador o encargado del departamento puede revisar las ofertas propias de su departamento y además las solicitudes realizadas a cada una de las ofertas. Podrá marcarlas como candidatas, aceptadas o rechazadas a través de la revisión del currículum y los datos del aspirante. También puede hacer una búsqueda directa en la bolsa del sitio, que no es más que un listado de los currículums activos, donde podrá filtrar según los criterios deseados; pudiendo establecer diferentes niveles para la competencia de los que aspiran a una plaza. Si un aspirante es candidato o resulta del interés del aprobador, este podrá enviar mensajes directos al aspirante con el fin de que el mismo establezca contacto con el departamento, debido a que este no podrá iniciar una conversación con el aprobador a través del sitio.



Figura 1 Propuesta del Sistema.

2.2 Características no funcionales del sistema.

2.2.1 Usabilidad

- Es importante que se posea una preparación previa para trabajar con el sistema.
- Los usuarios cuentan con una página estática dentro del sitio que les sirve de ayuda explicándole las funcionalidades u opciones en cada apartado

2.2.2 Disponibilidad

- Es importante y necesaria la ejecución de la aplicación las veinticuatro horas del día.
- La base de dato se actualizará en tiempo real.

2.2.3 Hardware

Los requerimientos para que funcione correctamente la aplicación son mínimos. Quedando fijos a los requerimientos básicos del servidor web y al servidor de base de datos. Aun así se ofrecen un conjunto de características básicas para la selección de la tecnología a utilizar en cuanto al hardware.

Para el correcto funcionamiento de los componentes del lado del servidor se hace necesario contar los siguientes elementos:

- 1 PC Intel Celeron a 2.6 GHz, 2 GB de memoria RAM, 150 GB de capacidad de disco duro para el servidor de base de datos y con 2 GHz como mínimo de velocidad de trabajo del procesador en el servicio.

CAPÍTULO 2 CARACTERÍSTICAS, EXPLORACIÓN Y PROPUESTA DEL SISTEMA

- 1 PC Intel Celeron a 2.6 GHz, 2 GB de memoria RAM, 80 GB de capacidad de disco duro para el proveedor de servicios y con 2 GHz como mínimo de velocidad de trabajo del procesador en el servicio.

Para las PC en puestos de trabajos situados en el lado del cliente, deberán contar al menos:

- 1 PC con las características básicas para navegar en la web.

2.2.4 Software

- Se requiere un navegador Web (Mozilla Firefox, Google Chrome, Safari, Opera o Internet Explorer) en las estaciones de trabajo.
- Las estaciones de trabajo clientes deberán utilizar cualquier sistema operativo de la familia de Windows superior a XP Service Pack 2 o un sistema operativo GNU/Linux.
- Se utilizará PostgreSQL versión 8.3 en el servidor.
- Para desplegar la aplicación se requiere del servidor HTTP Apache, el lenguaje Python y el framework Django.

2.2.5 Interfaz de usuario

- La aplicación propuesta poseerá una interfaz sencilla dirigida a las personas que se relacionen con el sistema.
- Tanto la interfaz, como los mensajes de interacción con el usuario serán en idioma español.

2.2.6 Soporte

- Una vez terminado el sistema se realizará proceso de despliegue, capacitación y mantenimiento de software.
- El personal que desarrolla el sistema debe contar con el nivel técnico requerido.

2.3 Personas relacionadas con el sistema

Se define como persona relacionada con el sistema al personal autorizado en el Departamento de Capital Humano y en cada uno de los departamentos subordinados a él:

- **Administrador:** usuario que posee todos los privilegios del sistema.
- **Asistente (encargado del sistema por el (CH)):** usuario que posee los privilegios referentes a todos los departamentos.
- **Aprobador:** usuario con permisos limitados para operar con la información referente a su departamento.
- **Aspirante:** usuario que representa al personal que solicita una plaza seleccionada.

2.4 Fase de Exploración

La metodología XP propone como primera fase la de Exploración, es aquí donde se define el alcance real del sistema, permitiendo una familiarización del equipo de desarrollo con las herramientas, tecnologías y procesos. Esta fase comienza por la creación de una serie de HU, las cuales definen mediante su redacción qué es lo que verdaderamente necesita el cliente.

2.4.1 Historias de Usuarios (HU)

Las HU representan una breve descripción del comportamiento del sistema, emplea terminología del cliente sin lenguaje técnico, solamente proporcionarán una breve descripción de las mismas, detalles sobre la estimación del riesgo y cuánto tiempo será empleado en su implementación, se realiza una por cada funcionalidad del sistema. Las HU permiten responder rápidamente a los requisitos cambiantes (39).

Dentro de la metodología XP las HU deben ser escritas por los clientes. Es el cliente el encargado de asignarle una prioridad a cada HU y es el equipo de desarrollo el encargado de asignarle un costo, este se traduce en las semanas que llevará el desarrollo de las mismas. Si las historias según lo planificado demoran en desarrollarse se sugiere dividirla en historias más pequeñas. También, es importante destacar, que las HU nuevas pueden describirse en cualquier momento, con esto se comprueba la flexibilidad de la metodología (39).

Clasificación de las Historias de Usuarios

El nivel de prioridad se definió teniendo en cuenta la precedencia de las funcionalidades. Para realizar algunas funcionalidades, otras tienen que estar previamente implementadas, por ejemplo, el poder

CAPÍTULO 2 CARACTERÍSTICAS, EXPLORACIÓN Y PROPUESTA DEL SISTEMA

registrarse en el sistema, es condición esencial para realizar cualquier otra operación. De igual manera se tuvo en cuenta la complejidad de las funcionalidades, asignándoles un mayor grado de importancia a las más complejas.

La Prioridad en el negocio:

Alta: Se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.

Media: Se le otorga a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.

Baja: Se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tiene nada que ver con el sistema en desarrollo.

El riesgo en el desarrollo:

Alta: Cuando en la implementación de las HU se consideran la posible existencia de errores que lleven a la inoperatividad del código.

Media: Cuando pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión.

Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Las HU serán representadas mediante tablas divididas por las siguientes secciones:

- **Número:** número de la historia de usuario incremental en el tiempo.
- **Nombre de Historia de Usuario:** el nombre de la historia de usuario sería para identificarlas mejor entre los desarrolladores y el cliente.
- **Usuario:** involucrados en la ejecución de la HU.
- **Iteración Asignada:** número de la iteración.
- **Prioridad en el Negocio:** Alta, Media o Baja.

CAPÍTULO 2 CARACTERÍSTICAS, EXPLORACIÓN Y PROPUESTA DEL SISTEMA

- **Riesgo en Desarrollo:** Alta, Media o Baja.
- **Programador Responsable:** involucrados en el desarrollo de la HU.
- **Puntos Estimados:** tiempo estimado que se demorará el desarrollo de la HU, debe ser menor que tres semanas.
- **Descripción:** breve descripción de la HU.
- **Observaciones:** señalamiento o advertencia del sistema.
- **Prototipo de Interfaz:** Prototipo de interfaz si aplica.

El cliente y el equipo de desarrollo trabajan en conjunto para definir cómo agrupar las HU para su lanzamiento. A continuación se muestran las HU de alta prioridad definida para la realización del sistema:

Tabla 3 HU # 1: Gestionar Usuario

Historia de usuario	
Número: 1	Usuario: usuario
Nombre de la Historia de Usuario: Gestionar Usuario	
Prioridad en el negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1\3	Iteración asignada: 1
Programador responsable: Jorge Ameth García Diago Melquiades Megret Ramírez	
Descripción: Permite registrarse en la aplicación como un nuevo usuario y después poder autenticarse usando su nombre y contraseña.	
Observaciones:	



Prototipo de interfaz:



Tabla 4 HU # 2: Enviar Notificaciones a los Aspirantes Aprobados

Historia de usuario	
Número: 2	Usuario: Aprobador
Nombre de la Historia de Usuario: Enviar Notificaciones a los Aspirantes Aprobados	
Prioridad en el negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1/3	Iteración asignada: 2

CAPÍTULO 2 CARACTERÍSTICAS, EXPLORACIÓN Y PROPUESTA DEL SISTEMA



Programador responsable: Jorge Ameth García Diago

Melquiades Megret Ramírez

Descripción: Permite al aprobador mostrar un mensaje al usuario informándole que ha sido aprobada su solicitud.

Observaciones:

Prototipo de interfaz:

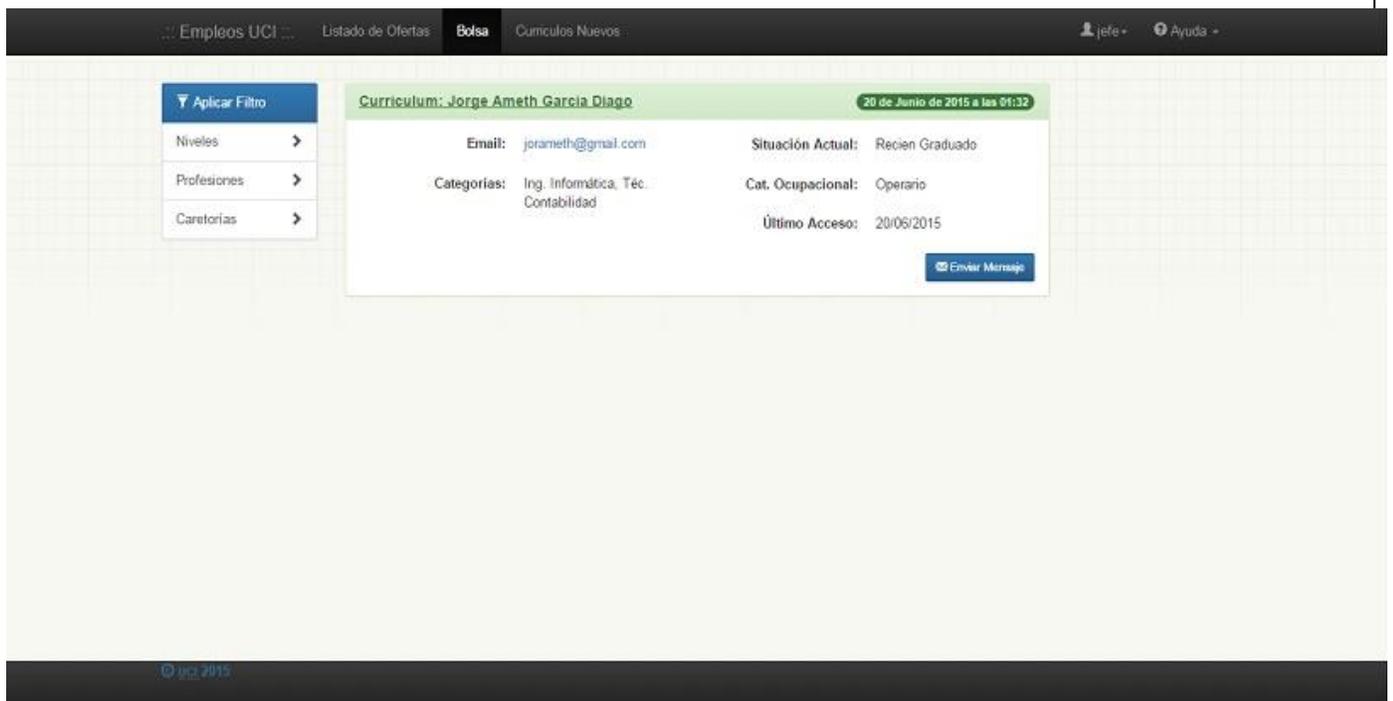
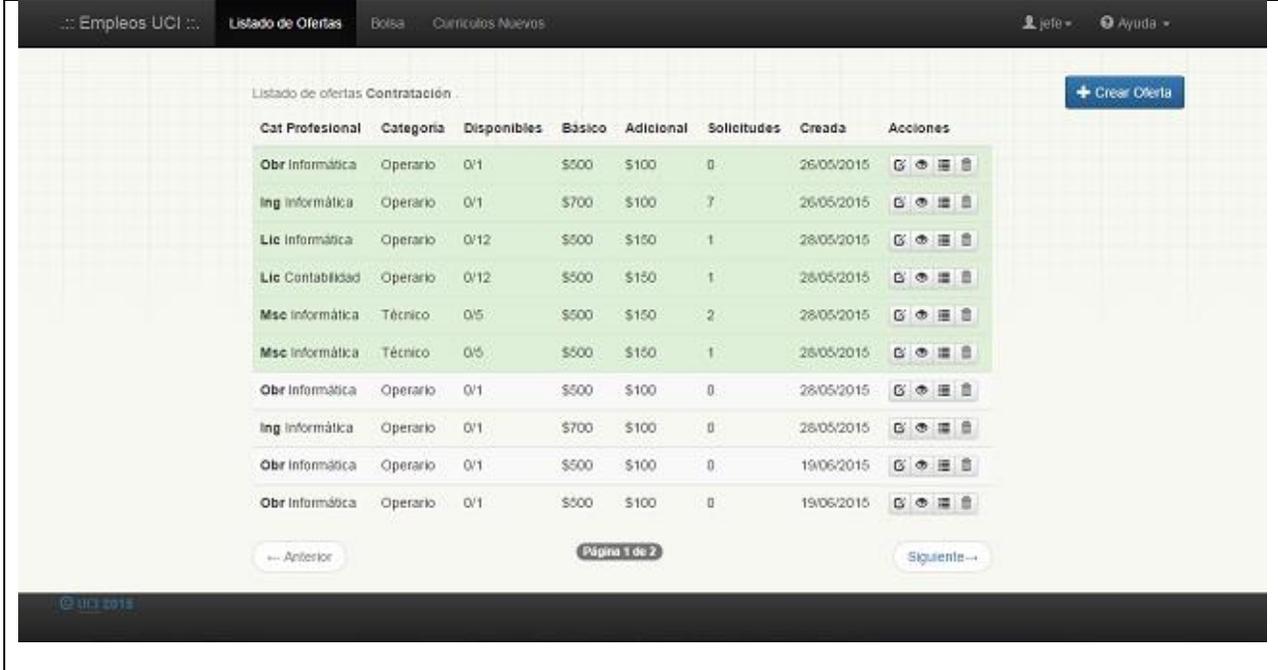


Tabla 5 HU # 3: Gestionar Ofertas

CAPÍTULO 2 CARACTERÍSTICAS, EXPLORACIÓN Y PROPUESTA DEL SISTEMA



Historia de usuario	
Número: 3	Usuario: Aprobador
Nombre de la Historia de Usuario: Gestionar Ofertas	
Prioridad en el negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1\3	Iteración asignada:
Programador responsable: Jorge Ameth García Diago Melquiades Megret Ramírez	
Descripción: Permite al Aprobador gestionar la información en cuanto a las ofertas.	
Observaciones:	
Prototipo de interfaz:	



... Empleos UCI ... Listado de Ofertas Bolsa Currículos Nuevos jefe Ayuda

Listado de ofertas Contratación

Cat Profesional	Categoría	Disponibles	Básico	Adicional	Solicitudes	Creada	Acciones
Obr Informática	Operario	0/1	\$500	\$100	0	26/05/2015	[G] [E] [D] [I]
Ing Informática	Operario	0/1	\$700	\$100	7	26/05/2015	[G] [E] [D] [I]
Lic Informática	Operario	0/12	\$500	\$150	1	28/05/2015	[G] [E] [D] [I]
Lic Contabilidad	Operario	0/12	\$500	\$150	1	28/05/2015	[G] [E] [D] [I]
Msc Informática	Técnico	0/5	\$500	\$150	2	28/05/2015	[G] [E] [D] [I]
Msc Informática	Técnico	0/5	\$500	\$150	1	28/05/2015	[G] [E] [D] [I]
Obr Informática	Operario	0/1	\$500	\$100	0	28/05/2015	[G] [E] [D] [I]
Ing Informática	Operario	0/1	\$700	\$100	0	28/05/2015	[G] [E] [D] [I]
Obr Informática	Operario	0/1	\$500	\$100	0	19/06/2015	[G] [E] [D] [I]
Obr Informática	Operario	0/1	\$500	\$100	0	19/06/2015	[G] [E] [D] [I]

... Anterior Página 1 de 2 Siguiente ...

© UCI 2015

Las demás HU podrán encontrarse en el [Anexo 1](#).

2.5 Planificación

La actividad de planeación en la metodología XP comienza con la creación de una serie de HU que describen las características y funcionalidades requeridas para el software a construir. Durante la fase de planeación se realiza una estimación del esfuerzo que costará implementar cada HU. Este se expresa utilizando como medida el punto.

Un punto se considera como una semana ideal de trabajo donde los miembros del equipo de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Esta estimación incluye todo el esfuerzo asociado a la implementación de la HU.

2.5.1 Estimación de esfuerzo por Historias de Usuario

A continuación se muestra la estimación del esfuerzo por cada HU propuesta para el desarrollo de la aplicación:

Tabla 6 Estimación de esfuerzo por Historias de Usuario

CAPÍTULO 2 CARACTERÍSTICAS, EXPLORACIÓN Y PROPUESTA DEL SISTEMA

Historia de Usuarios	Puntos de Estimación
1. Gestionar Usuario	1.6
2. Enviar Notificaciones a los Aspirantes Aprobados	0.4
3. Gestionar Ofertas	4
4. Completar Datos Personales	0.4
5. Completar otros Datos	0.6
6. Gestionar Solicitudes por Departamentos	3.4
7. Filtrar Ofertas por Departamentos	1
8. Revisar Mensajes y Notificaciones	0.4
9. Realizar Contratación	1

2.5.2 Plan de Iteraciones

Después de identificar, describir y estimar el esfuerzo de las HU dedicado a la realización de cada una de ellas, se procede a la planificación de la fase de implementación estableciendo una división de tres iteraciones.

2.5.2.1 Iteración 1

En esta iteración se lleva a cabo el desarrollo de las HU 1, 2 y 3. Esto representa que al terminar la iteración se ha implementado un 34% de la aplicación.

2.5.2.2 Iteración 2

CAPÍTULO 2 CARACTERÍSTICAS, EXPLORACIÓN Y PROPUESTA DEL SISTEMA

En esta iteración se lleva a cabo el desarrollo de las HU del número 4 al número 7. Esto representa que al terminar la iteración se ha implementado un 40% de la aplicación.

2.5.2.3 Iteración 3

En esta iteración se lleva a cabo el desarrollo de las HU 8 y 9. Esto representa que al terminar la iteración se ha implementado un 26% de la aplicación. Al terminar esta iteración se obtendrá una versión 1.0 del producto final y a partir de aquí el sistema se pondrá en función para ser evaluado.

2.5.3 Plan de Duración de las Iteraciones

El plan de duración de las iteraciones se encarga de mostrar las HU en el orden en que se implementarán en cada iteración así como la duración estimada de las mismas.

Tabla 7 Plan de duración de las iteraciones.

Iteraciones	Orden las HU a implementar	Duración Total
1	Gestionar Usuarios Enviar Notificaciones a los Aspirantes Aprobados Gestionar Ofertas	6 semanas
2	Completar otros Datos Personales Completar otros Datos Gestionar Solicitudes por Departamentos Filtrar Ofertas por Departamentos	5 semanas y 2 días
3	Revisar Mensajes y Notificaciones Realizar Contratación	1 semana y 2 día
		Total: 12 semanas y 4 días

2.5.4 Plan de Entrega

El plan de entrega detalla la fecha fin de cada iteración, así como en este caso las HU a las cuales se les dieron cumplimiento en cada iteración.

Tabla 8 Plan de Entrega.

Iteraciones	Historias de Usuarios				Fecha
Iteración 1	1	2	3		03-04-2015
Iteración 2	4	5	6	7	13-05-2015
Iteración 3	8	9			22-05-2015

2.6 Conclusiones

En este capítulo se describió la propuesta del sistema a desarrollar. Se identificaron las funcionalidades y requisitos que el sistema debe cumplir, así como la descripción de las HU divididas por iteraciones y la planificación del esfuerzo dedicado a la realización de cada una de ellas en el orden en que se les dará cumplimiento según las necesidades del cliente.

Capítulo 3. Diseño del Sistema

3 Introducción

En este capítulo se describe la fase de diseño, como parte de la metodología de desarrollo XP. Se identifican y organizan las clases relevantes para las funcionalidades del sistema así como el patrón arquitectónico utilizado para la aplicación web. También se procederá a realizar el diseño de la base de datos.

3.1 Arquitectura

La arquitectura del software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos, el contexto en el que se implantarán y los principios que orientan su diseño y evolución. Su objetivo principal es aportar elementos que ayuden a la toma de decisiones y al mismo tiempo, proporcionar conceptos y un lenguaje común que permitan la comunicación entre los equipos que participen en un proyecto (40).

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes (41).

Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos. Los clientes son estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red.

Una representación gráfica de este tipo de arquitectura sería la siguiente:

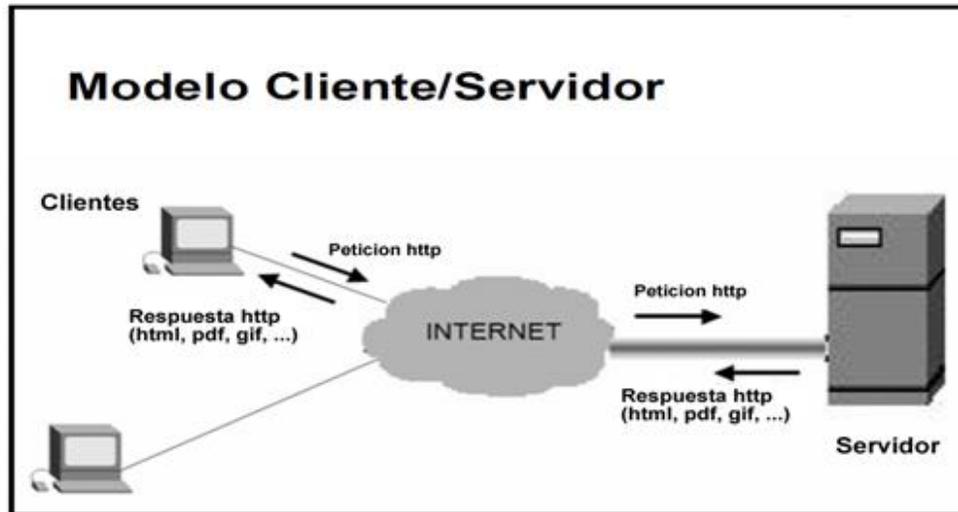


Figura 3 Arquitectura cliente servidor

3.2 Patrones Arquitectónicos

El desarrollo de software ha adquirido un notable auge trayendo consigo el aumento de la complejidad de los mismos. El uso de los patrones arquitectónicos es una solución a los problemas comunes que se presentan durante el desarrollo de sistemas de software, debido a que dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Los patrones de arquitectura expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos (42).

Para el desarrollo de este sistema el patrón de arquitectura que se seguirá es el Modelo Vista Plantilla (MTV por sus siglas en inglés) para la aplicación web.

3.2.1 Patrón Arquitectónico Modelo Vista Plantilla

En Django el patrón llamado Modelo Vista Plantilla (MTV por sus siglas en inglés) es sencillo de entender, el modelo sigue siendo el modelo, en este caso, la vista no es una vista, sino que más bien es un controlador que se llama vista, y las plantillas son las vistas del MVC, es decir, los formularios van en la plantilla, los formularios hacen peticiones a las vistas y las vistas obtienen datos de los modelos (43).

- **Modelo:** define los datos almacenados, se encuentra en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros, posee métodos también. Todo esto permite indicar y controlar el comportamiento de los datos.
- **Vista:** se presenta en forma de funciones en Python, su propósito será determinar qué datos serán visualizados, entre otras cosas más. La vista también se encarga de tareas conocidas como el envío de correo electrónico, la autenticación con servicios externos y la validación de datos a través de formularios. Lo más importante a entender con respecto a la vista es que no tiene nada que ver con el estilo de presentación de los datos, sólo se encarga de los datos, la presentación es tarea de la plantilla.
- **Plantilla:** es básicamente una página HTML con algunas etiquetas extras propias de Django, en sí no solamente crea contenido en HTML (también XML, CSS, JavaScript, CSV, etc.). Esta recibe los datos de la vista y luego los organiza para la presentación al navegador web. Las etiquetas que Django usa para las plantillas, permiten que sea flexible para los diseñadores de la misma, incluso tiene estructuras de datos como if, por si es necesaria una presentación lógica de los datos, estas estructuras son limitadas para evitar un desorden poniendo cualquier tipo de código Python.

Una representación gráfica de este tipo de arquitectura aplicado al sistema sería la siguiente:

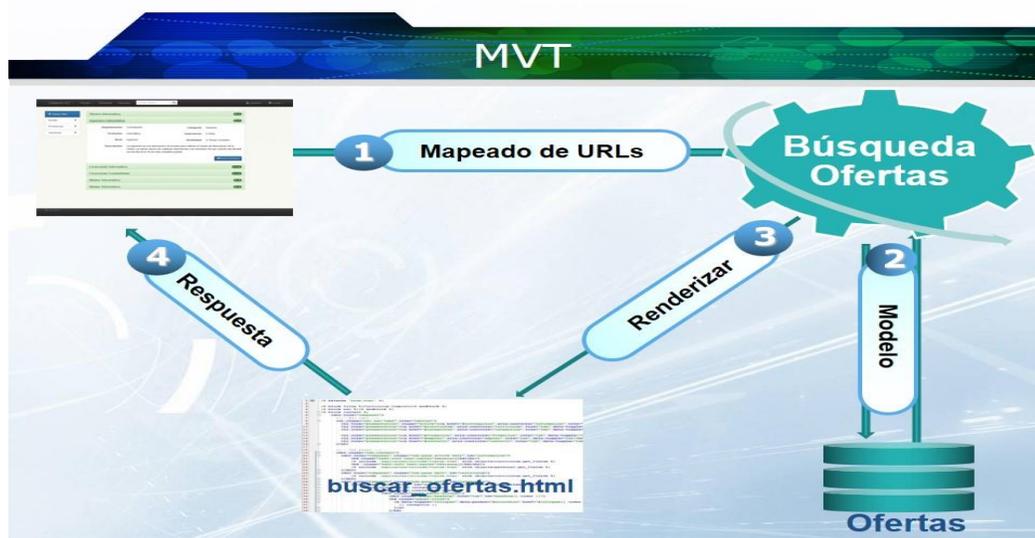


Figura 4 Arquitectura Modelo Vista Plantilla.

En la figura anterior se muestran los pasos que se siguen en la interacción de cada una de las capas del modelo vista plantilla:

1. El navegador envía una solicitud y se realiza un mapeado de las URLs para ubicar la vista apropiada.
2. La vista interactúa con el modelo para obtener los datos.
3. La vista llama a la plantilla para renderizar resultados.
4. La plantilla renderiza la respuesta a la solicitud del navegador.

3.3 Diagrama de Clases

A continuación se presentan los diagramas de clases divididos por capas de la arquitectura MVT para un mejor entendimiento de funcionamiento del framework Django basado en esta arquitectura para el servicio de contratación a desarrollar.

3.3.1 Diagrama de Clases de la Aplicación Web

Capa Modelo

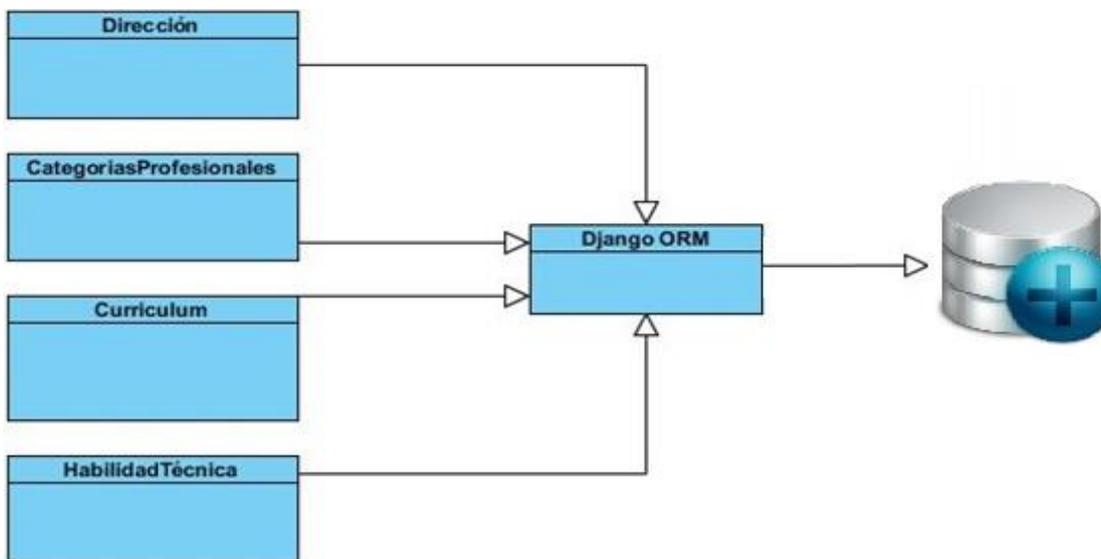


Figura 6 Capa de modelo

Dirección: Gestiona la dirección de los usuarios.

CategoriasProfesional: Se encarga de la gestión del nivel profesional de los usuarios.

Curriculum: Aquí estará toda la configuración relacionada con el usuario.

HabilidadTécnica: Gestiona el dominio de las técnicas del usuario.

DjangoORM: Este es el ORM que se utiliza para el trabajo con la base de datos.

Capa Vista

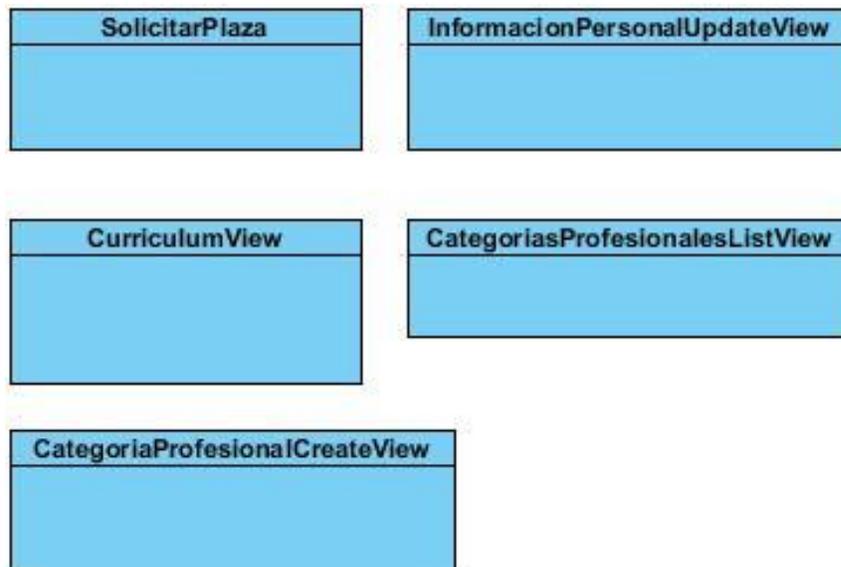


Figura 7 Capa vista

SolicitarPlazas: Maneja y controla el flujo de acciones de la información en cuanto a la solicitud de plazas vacantes en la entidad.

InformaciónPersonalUpdateView: Se encarga de la actualización de la información del personal.

CurriculumView: Maneja y controla el flujo de acciones de los currículos.

CategoriaProfesionalesCreateView: Tiene la información para la creación de una nueva categoría profesional.

CategoriaProfesionalesListView: Maneja la información para devolver los formularios para la edición de las categorías profesionales.

Capa Plantilla

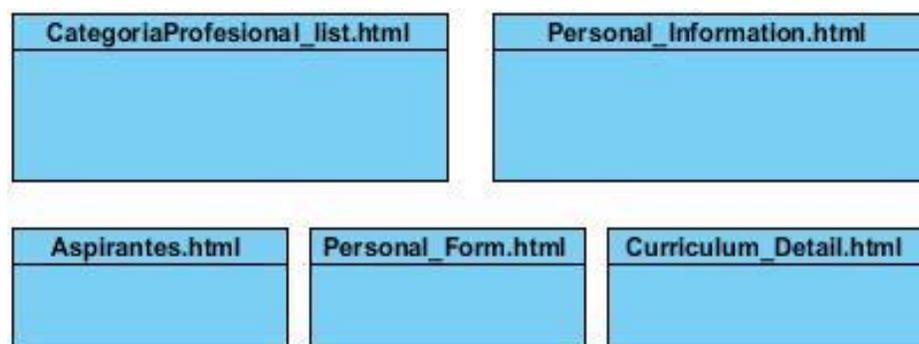


Figura 8 Capa plantilla

CategoríaProfesional_list.html, Personal_Information.html, Aspirantes.html, Personal_Form.html y Curriculum_Detail.html estos archivos se encuentran dentro de la carpeta templates del módulo al cual pertenecen, la cuales definen el modelo plantilla del patrón arquitectónico MVT en el framework Django. Los mismos definen las plantillas HTML.

3.4 Patrones de Diseño

Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan (44).

Características

- Contribuyen a dar flexibilidad y extensibilidad a los diseños.
- Contribuyen a reutilizar diseño, identificando aspectos claves de la estructura de un diseño que puede ser aplicado en una gran cantidad de situaciones.
- Un proyecto o estructura de implementación que logra una finalidad determinada.

- Mejoran (aumentan, elevan) la flexibilidad, modularidad y extensibilidad. Incrementan el vocabulario de diseño, ayudando a diseñar desde un mayor nivel de abstracción.
- Es una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios.
- Es un lenguaje de programación de alto nivel.

3.4.1 Patrones para Asignar Responsabilidades (GRASP)

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades.

Para el desarrollo de este sistema se utilizaron los siguientes patrones GRASP:

Experto

Problema: ¿Cómo lograr que cada clase cumpla con la responsabilidad que le corresponde?

Solución: Asignar la responsabilidad a la clase que tiene la información necesaria para cumplir con la responsabilidad.

Ejemplo: Las clases **SolicitudesList** y **Curriculum** en el módulo vistas de la aplicación aspirante cuenta con la información necesaria para cumplir cada una de las responsabilidades que les corresponden.

Creador

Problema: ¿Qué clase debería ser la responsable de crear una nueva instancia de alguna clase?

Solución: Asignarle a una clase la responsabilidad de crear una instancia necesaria de otra clase.

Ejemplo: Todas las clases que heredan de la clase genérica **CreateView** devuelve el formulario que facilita la edición de él, o los modelos especificados en las clases especializada. Finalizando con la creación de una nueva instancia del modelo o la devolución formulario con los errores del correspondientes que el usuario podría corregir y reenviar nuevamente para su validación y procesado.

Controlador

Problema: ¿Quién debería encargarse de atender un evento del sistema?

Solución: Un Controlador es un objeto de interfaz no destinada al usuario que se encarga de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una o varias clases.

Ejemplo: La clase **View** despacha las peticiones realizadas al servidor provenientes del cliente. Según el tipo de la petición (get, post, put), las vistas definidas en el sistema que especializan esta clase genérica, actúan como controladores que responden a la petición del cliente según sea el tipo de evento.

Alta Cohesión

Problema: ¿Cómo mantener la complejidad dentro de los límites manejables?

Solución: Este patrón es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

Ejemplo: A las clases **SolicitudesList** y **CategoriaProfesionalUpdateView** se les asignan responsabilidades con el objetivo de que trabajen en la misma área de aplicación y que no tengan mucha complejidad.

Bajo Acoplamiento

Problema: ¿Cómo lograr una baja dependencia y una alta reutilización de clases?

Solución: Asignar responsabilidades manteniendo bajo el acoplamiento.

Ejemplo: A las clases **CategoriaProfesionalCreateView**, **CategoríaProfesionalUpdateView** se le asignan responsabilidades de forma tal que solo se comuniquen con la clase **CategoriaprofesionalForm** la cual se encarga de procesar los datos del formulario y crear la nueva instancia.

3.4.2 Patrones de diseño (GOF)

Creacionales: Solucionan problemas de creación de instancias. Ayudan a encapsular y abstraer dicha creación.

Singleton

Problema: ¿Cómo lograr tener una clase que sólo tenga una instancia y proporcione un punto de acceso global a ésta instancia?

Solución: La solución es crear un método dentro de la clase que instancie a la propia clase si esta ya no está instanciada.

Ejemplo: La clase **setting.py** implementa este patrón ya que se instancian una sola vez y permiten el acceso global a ella.

3.5 Tarjeta Clase-Responsabilidad-Colaborador

La metodología de desarrollo XP como parte de la fase de diseño propone el modelado de Clase-Responsabilidad-Colaborador (CRC), lo que constituye un modelo simple de organizar las clases más relevantes para las funcionalidades del sistema. Este modelado CRC utiliza tarjetas, con el objetivo de desarrollar una representación organizada de las clases.

Un modelo CRC es en realidad una colección de tarjetas índices estándar que representan clases.

Las tarjetas se dividen en tres secciones:

- A lo largo del borde superior de la tarjeta se escribe el nombre de la clase.
- En el cuerpo de la tarjeta a la izquierda se listan las responsabilidades de la clase.
- En el cuerpo de la tarjeta a la derecha los colaboradores.

Una **responsabilidad** es una acción que la clase sabe o hace. Los **colaboradores** son aquellas clases que se requieren para que una clase reciba la información necesaria para completar una responsabilidad.

A continuación las tarjetas CRC correspondientes a las clases más relevantes para las funcionalidades del servicio de contratación.

Tabla 9 CRC Clase SolicitarPlazas

Clase SolicitarPlazas
Descripción: Clase encargada de solicitar las plazas ofertadas por los departamentos



Responsabilidad	Colaborador
Solicitar plazas ofertadas por los departamentos	InformaciónPersonalUpdateView

Tabla 10 CRC Clase CategoriaProfesionalCreateView

Clase CategoriaProfesionalCreateView	
Descripción: Permite llenar los formularios para la edición de las categorías profesionales.	
Responsabilidad	Colaborador
Guarda los datos de la nueva categoría profesional	CategoriaProfesionalForm
Chequea si el usuario no repite la misma categoría	CategoríaProfesional

Tabla 11 CRC Clase InformaciónPersonalUpdateView

Clase InformaciónPersonalUpdateView	
Descripción: Clase encargada de permitir el ingreso de la información personal del usuario.	
Responsabilidad	Colaborador
Guarda los datos personales	PersonalForm

Las demás tarjetas las podrán encontrar en el [Anexo 2](#).

3.6 Diagrama de clases persistentes

Las clases persistentes son aquellas que tienen durabilidad en el tiempo. Son el punto de partida para la creación del modelo físico de datos.

A continuación se muestran las clases principales del diagrama de clases persistentes correspondiente al proceso de contratación de la UCI.

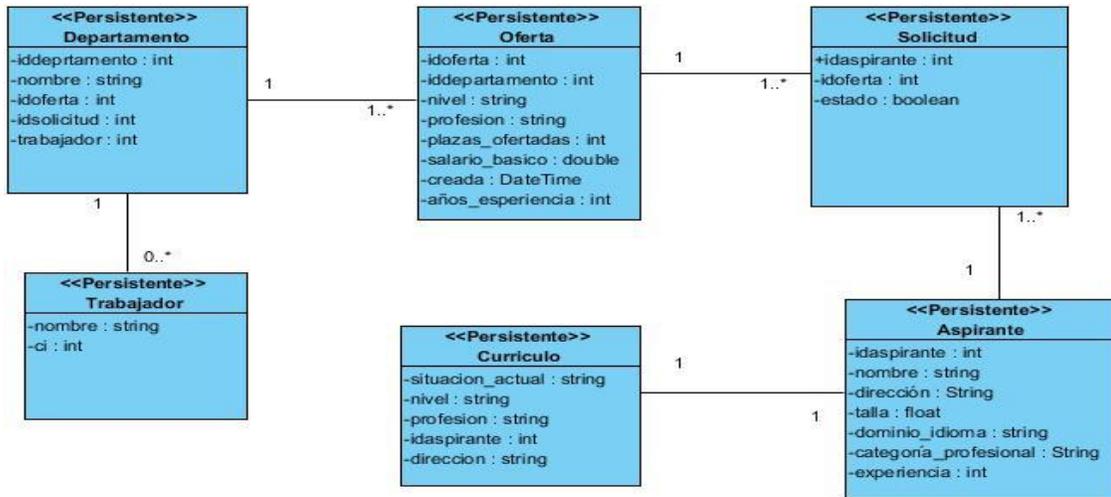


Figura 9 Diagrama de clases persistentes

3.7 Modelo Físico de Base de Datos

Partiendo del diagrama de clases persistente visto anteriormente, se define el siguiente modelo físico de la base de datos:

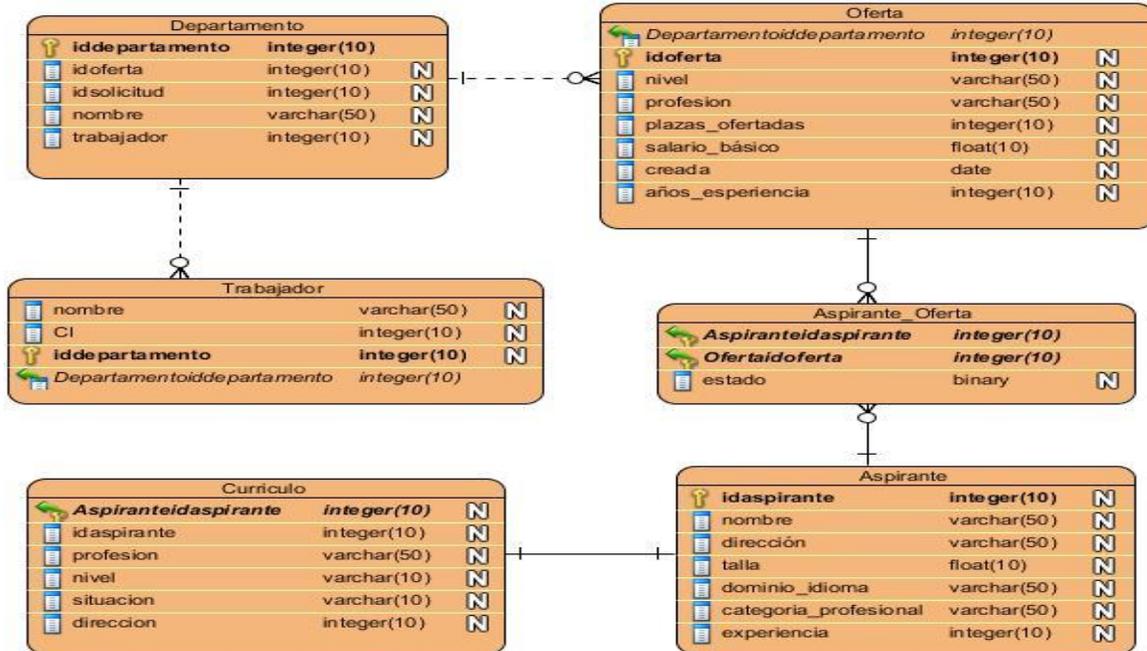


Figura 10 Modelo Físico de la base de datos

3.8 Conclusiones

En este capítulo se realizó un análisis sobre el patrón arquitectónico y de diseño utilizado por el framework sobre el que se desarrolló la aplicación, así como se propone la arquitectura a utilizar en el desarrollo del sistema. Se identificaron y describieron las tareas de la ingeniería listas para su implementación, correspondiente a las HU descritas en el capítulo anterior. Se identificaron las clases que persisten en el tiempo para obtener a partir de las mismas el modelo físico de la base de datos y se realizan las tarjetas CRC como elemento importante en la descripción de cada una de las clases para la posterior implementación del sistema.

Capítulo 4 Implementación y Prueba

4 Introducción

En este capítulo se describen las fases de implementación y prueba, propias de todas las historias de usuarios que se definieron anteriormente como parte de la metodología XP. Se le realizarán las pruebas al software, las cuales se derivan de las Historias de Usuario y Tareas de la Ingeniería que se han implementado como parte del lanzamiento del software.

Para la implementación de un software, XP plantea hacer varias iteraciones, obteniendo al final de cada una un producto final, que debe ser probado y mostrado al cliente. En el transcurso de cada iteración, se descomponen la HU definidas por el cliente en fase de Exploración, en tareas de la ingeniería que entregan a los programadores para ser implementadas durante la iteración correspondiente.

4.1 Estándares de Codificación

Con la utilización de XP es necesario seguir ciertos estándares de programación ya que este enfatiza más la parte relativa del código. De acuerdo con el tipo de lenguaje que utilizan y las normas que propone el mismo, los programadores lo realizan para trabajar de forma coordinada dentro del proyecto. Estos mantienen el código legible para los miembros del equipo facilitando los cambios y su mantenimiento (45).

A continuación se describen algunos estándares utilizados en la implementación de la propuesta de solución.

4.1.1 Formato de Código

- **Identación:** Usar cuatro espacios por cada nivel de indentación. Para código realmente antiguo que no se quiera estropear, se puede continuar usando tabuladores de 8 espacios.
- **Tabuladores o espacios:** No se mezclan tabuladores y espacios.
- **Longitud de línea:** Limita todas las líneas a un máximo de 79 caracteres.

- **Líneas en Blanco:** Se separan las funciones no anidadas y las definiciones de clases con dos líneas en blanco. Las definiciones de métodos dentro de una misma clase se separan con una línea en blanco. Usar líneas en blanco dentro de las funciones, para indicar secciones lógicas.
- **Importaciones:** Normalmente se colocan en distintas líneas siempre que sea posible. Siempre en la parte superior del archivo después de cualquier comentario y antes de las variables globales y las constantes de modulo. Después de cada grupo de importación se añade una línea en blanco.
- **Espacios en blanco:** No usar espacio alrededor del signo de igualdad, cuando se use para indicar el nombre de un argumento o el valor de un parámetro por defecto, ni antes de salir del paréntesis o después de entrar al paréntesis. Usar alrededor de los operadores aritméticos e inmediatamente antes de una coma, punto y coma, corchetes, llaves o dos puntos.
- **Comentarios:** Lograr que constituya una frase completa. Los que están en bloque generalmente se aplican al código que se encuentra a continuación al mismo nivel que dicho comentario. Los de líneas comienzan por un símbolo de número y un espacio y se encuentran separados por dos espacios como mínimo de la sentencia.
- **Nombres:** Las convenciones actuales para el nombre varían de una fuente a otra. No es posible lograr una consistencia en su totalidad, pero sí se pueden aplicar diferentes estilos que resultan de mucha utilidad.

Descriptivos: Estilo de Nombrado: B (una sola letra mayúscula), b (una sola letra minúscula), minúsculas, minúsculas_con_guiones_bajos, MAYUSCULAS, MAYUSCULAS_CON_GUIONES_BAJOS.

Prescriptivo: Convenciones de Nombrado.

- **Nombres a evitar:** Evitar usar los caracteres 'O', 'l' o 'i' como nombres de variables de un solo carácter.
- **Nombres de paquetes y módulos:** Seleccionar nombres cortos en letras minúsculas. Se puede utilizar guiones bajos si mejora la legibilidad.
- **Nombres de clases:** Generalmente, los nombres de clases usan la convención CapWords. Las clases de uso interno tienen además un guión bajo al principio del nombre.

- **Nombres de excepciones:** Dado que las excepciones deberían ser clases, es aconsejable aplicar la convención relativa al nombrado de clase. Usar el sufijo 'Error' en los nombres de las excepciones que sean realmente un error.
- **Recomendaciones para la programación:** Las comparaciones con None siempre deben realizarse con 'is' o 'is not', evitando los operadores de igualdad. Es importante no escribir "if x" cuando realmente se desea evaluar "if x is not None". Utiliza excepciones basadas en clases. Los módulos o paquetes deberían definir sus propias clases excepción específica para el dominio del problema, las cuales deberían heredar de la clase Excepción. No comparar valores booleanos con True o False usando ==.

4.2 Tareas de la Ingeniería

Las tareas de la ingeniería consisten en una descripción más detallada y orientada al programador de cada una de las funcionalidades con que cuenta el sistema para una correcta implementación de las mismas.

Las tareas de la ingeniería serán representadas mediante tablas divididas por las siguientes secciones:

- **Número Tarea:** los números deben ser consecutivos.
- **Número Historia de Usuario:** número de la historia de usuario a la que pertenece la tarea.
- **Nombre Tarea:** nombre que identifica a la tarea.
- **Tipo de Tarea:** las tareas pueden ser de: desarrollo, corrección, mejora u otra.
- **Puntos Estimados:** tiempo estimado en semanas que se le asignará a su desarrollo, no puede ser mayor que tres semanas.
- **Fecha Inicio:** fecha en que inicia el desarrollo de la tarea.
- **Fecha Fin:** fecha en que finaliza el desarrollo de la tarea.
- **Responsable:** nombre y apellidos del programador.
- **Descripción:** breve descripción de la tarea.

A continuación las tareas de la ingeniería correspondientes a las HU de prioridad más alta:

Tabla 12 Tarea # 1 Registrar Usuario



Tareas de la Ingeniería	
Número de Tarea: 1	Número de Historia de Usuario: 1
Nombre de Tarea: Registrar Usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 23-02-2015	Fecha Fin: 26-02-2015
Programador Responsable: Jorge Ameth García Diago Melquiades Megret Ramírez	
Descripción: Permite al usuario establecer un usuario y contraseña para así poder autenticarse después y acceder al sistema.	

Tabla 13 Tarea # 2 Autenticar Usuario

Tareas de la Ingeniería	
Número de Tarea: 2	Número de Historia de Usuario: 1
Nombre de Tarea: Autenticar Usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 27-02-2015	Fecha Fin: 04-03-2015
Programador Responsable: Jorge Ameth García Diago Melquiades Megret Ramírez	
Descripción: Permite al usuario autenticarse usando su nombre y contraseña.	



Tabla 14 Tarea # 3 Enviar Notificaciones a los Aspirantes Aprobados

Tareas de la Ingeniería	
Número de Tarea: 3	Número de Historia de Usuario: 2
Nombre de Tarea: Enviar Notificaciones a los Aspirantes Aprobados	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1/3
Fecha Inicio: 05-03-2015	Fecha Fin: 06-03-2015
Programador Responsable: Jorge Ameth García Diago Melquiades Megret Ramírez	
Descripción: Permite al aprobador mostrar un mensaje al usuario informándole que ha sido aprobada su solicitud.	

Las demás tareas se podrán encontrar en el [Anexo 3](#).

4.3 Pruebas

Dentro de los instrumentos capaces de medir el estado de calidad de un producto se encuentran las pruebas. El proceso de pruebas se dirige fundamentalmente a componentes del software o al sistema de software en general, con el objetivo de medir hasta donde el software cumple las funcionalidades establecidas por el cliente. Las pruebas del software verifican y revelan la calidad de un producto. Son utilizadas para identificar posibles errores en la implementación, calidad, o usabilidad de un programa de software.

Propio de la metodología XP, se lleva a cabo la fase de prueba. Durante el desarrollo de software, XP establece probar constantemente tanto como sea posible, esto permite un aumento de la calidad del sistema desarrollado reduciendo el número de errores no detectados.

XP divide las pruebas del sistema en dos grupos:

4.3.1 Pruebas Unitarias

Las pruebas unitarias son las encargadas de verificar el código y son diseñadas por los programadores. Cada uno de los desarrolladores tiene que ir probando constantemente lo que va obteniendo en el transcurso de la implementación de un sistema, para garantizar que las funcionalidades exigidas por el cliente estén implementadas correctamente. Estas fueron desarrolladas constantemente cada vez que se concluía la implementación de alguna funcionalidad probándola directamente en el entorno real. Todos los errores encontrados en las pruebas realizadas fueron corregidos (46).

Haciendo uso del PyCharm se logró llevar a cabo estas tareas, gracias a las facilidades que brinda este IDE en la realización de las pruebas unitarias, las cuales se realizaron sobre cada una de las funcionalidades implementadas. Para aplicar las pruebas, se generaron primero los resultados esperados. En la figura 9 se muestra un ejemplo del código creado en unas de las pruebas unitarias realizadas.

```

1  # coding=utf-8
2  from django.core.urlresolvers import reverse
3  from django.test import TestCase
4  from django.contrib.auth.models import User
5
6
7  class TestModels(TestCase):
8      def setUp(self):
9          User.objects.create_user(username='jorameth', email='jorameth@gmail.com', password='s3cr3t')
10         User.objects.create_user(username='meme', email='mmegret@estudiantes.uci.cu', password='s3cr3t')
11
12     def test_db(self):
13         self.assertEqual(User.objects.count(), 2)
14
15     def test_user_can_login(self):
16         username = 'jorameth' # Se busca al usuario y se prueba si existe
17         user = User.objects.get_by_natural_key(username)
18         self.assertIsNotNone(user.id)
19         self.client.login(username=username, password='s3cr3t') # Se autentica al usuario
20         self.assertEqual(user.is_authenticated(), True) # Se prueba nuevamente que que el usuario ya se autenticó
21
22
23     class TestViews(TestCase):
24     def test_home(self):
25         response = self.client.get(reverse('home'))
26         self.assertEqual(response.status_code, 200) # No se requiere que el usuario este autenticado
27
28     def test_ayuda(self):
29         response = self.client.get(reverse('empleos:ayuda'))
30         self.assertNotEqual(response.status_code, 200) # Se requiere que el usuario este autenticado

```

Figura 9 Código de unas de las pruebas realizadas

A continuación se muestra el resultado obtenido de la prueba anteriormente descrita. Se puede observar que se ejecutaron satisfactoriamente cuatro pruebas en un tiempo de 0.308 segundos.

Creating test database for alias 'default'...

Ran 4 tests in 0.308s

OK

Destroying test database for alias 'default'...

I:\Tesis\Proyecto\Codigo\empleosuci>

Las demás pruebas unitarias se podrán encontrar en el archivo test.py de cada uno de los módulos.

Después de haberle realizado pruebas unitarias a cada una de las funcionalidades exigidas por el cliente para la primera iteración se detectaron un total de 11 no conformidades, de las cuales:

- En la HU # 1 se detectaron dos no conformidades.
- En la HU # 3 se detectaron cuatro no conformidades.
- En la HU # 6 se detectaron dos no conformidades.
- En la HU # 9 se detectaron tres no conformidades.



Figura 10 resultado inicial de las pruebas unitarias.

Las no conformidades encontradas fueron resultas en una segunda iteración. Con el empleo de este tipo de pruebas se garantizó que las funcionalidades exigidas por el cliente se implementaran correctamente,

puesto que se comprobaron los resultados obtenidos al instante, disminuyendo el número de errores no detectados.

4.3.2 Prueba de Aceptación

Las pruebas de aceptación también llamadas pruebas del cliente las especifica el cliente y se enfocan en las características generales y las funcionalidades del sistema. En estas serán probadas las funcionalidades exigidas por el cliente, descritas en las HU que se han implementado (47).

Las pruebas de aceptación correspondientes a cada una de las funcionalidades del proceso de contratación representadas mediante tablas divididas por las siguientes secciones:

- **Código:** Identificador de la prueba y a la vez hará referencia al nombre de la misma.
- **Funcionalidad:** nombre de la funcionalidad de la tarea a la que hace referencia la prueba a realizar.
- **Nombre:** nombre de la prueba a realizar.
- **Descripción:** descripción de la prueba a realizar.
- **Condiciones de Ejecución:** condiciones que deben cumplirse para llevar a cabo la prueba.
- **Entradas / Pasos de Ejecución:** descripción de cada uno de los pasos a seguir en el desarrollo de la prueba, tiene en cuenta cada una de las entradas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.
- **Resultado Esperado:** breve descripción del resultado que se espera obtener con la prueba realizada.
- **Evaluación de la Prueba:** evaluación emitida sobre la prueba realizada. Esta evaluación tendrá uno de los tres resultados que a continuación se describen:
 - **Satisfactoria:** el resultado de la prueba es el esperado.
 - **Parcialmente Satisfactoria:** el resultado no es completamente el esperado por el cliente o usuario de la aplicación y mostrando resultados erróneos o fuera de contextos.

- **Insatisfactoria:** el resultado de la prueba esperada genera un error de codificación en la aplicación, se muestran elementos no deseados o fuera de contextos trayendo como consecuencia que la funcionalidad quede invalida.

Las pruebas de aceptación se llevarán a cabo redactando los casos de prueba, teniendo en cuenta el orden de las HU y la prioridad que ha sido asignada a las funcionalidades. Luego se hará la planificación con el cliente de cuándo y cuáles pruebas serán llevadas a cabo, para así reunir los miembros del proyecto seleccionados para realizarlas. Finalmente, se completarán cada uno de los campos de las tablas de las pruebas de aceptación con el resultado de la prueba (47).

A continuación se muestra un escenario para dichos casos de pruebas de aceptación realizados para la HU1, ellos son: Autenticar usuario correctamente y Autenticar usuario incorrectamente, modificándose el resultado de la prueba.

Tabla # 15 Prueba de Aceptación Autenticar Usuario

Prueba de Aceptación	
Código: HU1_CP1	Funcionalidad: 2
Nombre: Autenticar Usuario correctamente	
Descripción: Una vez ejecutada la aplicación se mostrará un formulario para introducir el usuario la contraseña. Si la persona no se encuentra registrada en el sistema no se podrá acceder al mismo, por otro lado en caso de estarlo y de ser correctos los valores introducidos, el sistema muestra la interfaz que le corresponde al usuario autenticado según el rol que posea.	
Condiciones de Ejecución: Que la persona esté registrada en el sistema.	
Entradas / Pasos de Ejecución: El usuario se autentica en el sistema.	
Resultado Esperado: Se espera que se muestre la interfaz correspondiente al rol del usuario.	
Evaluación de la Prueba: Satisfactoria	

Tabla # 16 Prueba de Aceptación Autenticar Usuario

Prueba de Aceptación	
Código: HU1_CP2	Funcionalidad: 2
Nombre: Autenticar Usuario incorrectamente	
Descripción: Se desea probar que la aplicación no permite la entrada de datos erróneos en el momento de la autenticación.	
Condiciones de Ejecución: Que la persona introduzca los datos incorrectos o incompletos.	
Entradas / Pasos de Ejecución: El usuario se autentica en el sistema.	
Resultado Esperado: La aplicación mostrará una notificación al usuario acerca de los datos erróneos o incorrectos para que vuelva a introducirlos.	
Evaluación de la Prueba: Satisfactoria	

Las demás pruebas de aceptación se podrán encontrar en el [Anexo 5](#).

Se realizaron dos iteraciones para estas pruebas.

En la primera iteración se detectaron un total de 7 no conformidades de las cuales:

- En la HU # 1 se detectaron tres no conformidades.
- EN la HU # 2 se detectó una no conformidad.
- En la HU # 6 se detectaron tres no conformidades.



Figura 9 Resultado inicial de las pruebas de aceptación.

Las no conformidades detectadas se resolvieron en la segunda iteración, donde el cliente especifico que los resultados de los casos de pruebas se correspondían con los resultados esperados.

Aplicadas las pruebas de aceptación, el cliente para el cual se desarrolló el sistema informático pudo comprobar el cumplimiento satisfactorio de cada una de las funcionalidades demandadas, quedando de esta forma la aplicación apta para su empleo.

4.4 Conclusiones

En el presente capítulo se describió el proceso de implementación del sistema detallando los patrones y arquitectura utilizados para el desarrollo del mismo. Se identificaron y describieron las tareas de la ingeniería listas para su implementación correspondientes a las HU descritas en el capítulo anterior. Se planteó a grandes rasgos los tipos de pruebas que se le realizaron al sistema y se elaboraron los casos de prueba de aceptación que se le harán al sistema, logrando la satisfacción del cliente con el software desarrollado.

Conclusiones Generales

En el presente trabajo se describieron todos los procesos que fueron identificados y que intervienen en el proceso de contratación de la UCI, obteniendo una mejor comprensión de cómo funciona el servicio desarrollado. Se seleccionaron las herramientas y la metodología para el desarrollo del sistema, así como la tecnología que interviene en el desarrollo de aplicación. Se realizó el estudio y comprensión del estilo arquitectónico así como de los patrones de diseño los cuales permitieron un mejor diseño y facilitaron la implementación del sistema. Se llevaron a cabo las pruebas del sistema en el transcurso de la implementación del mismo para corregir todos los posibles errores y fallas.

Con el desarrollo de esta Herramienta para la Gestión Empleadora del Departamento de Capital Humano de la Universidad de las Ciencias Informáticas, se logró dar solución a cada uno de los principales problemas presentes en el proceso de contratación.

Por todo lo anteriormente expuesto se concluye que se le ha podido dar solución a cada uno de los objetivos específicos y por tanto al objetivo general planteado logrando erradicar el problema a resolver propuesto al inicio del trabajo.

Recomendaciones

El equipo de desarrollo recomienda:

- Integrar el sistema desarrollado con el Protocolo Ligero/Simplificado de Acceso a Directorios (LDAP por sus siglas en inglés) de la UCI, para la gestión de los trabajadores de la Universidad.
- Extender el sistema con nuevas funcionalidades como consecuencia del surgimiento de nuevos requerimientos por parte del cliente.
- Actualizar el sistema a versiones superiores y de soporte extendido del lenguaje Python y el framework Django a medida que se vayan liberando nuevas versiones de estos.



Referencias Bibliográficas

1. **Jardinez, Garcia Dousat y Reyes.** *"Cuadro de mando integral para el control de gestión del proceso selección e integración al empleo del capital humano en una agencia empleadora"*. Cuba : Contribuciones a la Economía, 2010. pág. 2.
2. **Gérald Destinobles, A.** *El capital humano en las teorías del crecimiento económico*. Cuba : Edición electrónica gratuita, 2006.
3. **Garcia Dousat, Reyes Jardinez y Díaz Gamboa.** *"¿Cómo diagnosticar el Sistema de Gestión Integrada de Capital Humano?"*. Cuba : Observatorio de la Economía Latinoamericana, 2011. 148.
4. **Jardinez, Garcia Dousat y Reyes.** *"Cuadro de mando integral para el control de gestión del proceso selección e integración al empleo del capital humano en una agencia empleadora"*. Cuba : Contribuciones a la Economía, 2010. pág. 3.
5. **Reyes, Raul.** *Empleare. Empleare*. [En línea] Valencia, 4 de Noviembre de 2009. [Citado el: 5 de Diciembre de 2014.] <http://www.empleare.com/que-es-un-contrato-laboral.html>.
6. **E., Fernando Vidal.** *Guía para la Contratación*. Andalucía España : Consejería de Medio Ambiente, IDEAS, Bakeaz, 2006.
7. **Antunes, Rafael Obrego.** *Comrapublicaetica. Comrapublicaetica*. [En línea] Villafranca de Córdoba, ciudad Cordoba España, 13 de Diciembre de 2002. [Citado el: 19 de Diciembre de 2014.] <http://www.comrapublicaetica.org/index.php/le-ayudamos-paso-a-paso/proceso-de-contratacion..>

8. **Jardínez, Garcia Dousat y Reyes.** *"Cuadro de mando integral para el control de gestión del proceso selección e integración al empleo del capital humano en una agencia empleadora"*. Cuba : Contribuciones a la Economía, 2010. pág. 4. 5.
9. **Jardínez., MSc. Ing. Yordanis Garcia Dousat. Ing. Lixania Reyes.** *PROCEDIMIENTO PARA LA GESTIÓN DE SELECCIÓN E INTEGRACIÓN AL EMPLEO DEL CAPITAL HUMANO DESDE LA PERSPECTIVA DE UNA AGENCIA EMPLEADORA*. Valencia : Bekaez, 2006.
10. **Jardínez, Garcia Dousat y Reyes.** *"Cuadro de mando integral para el control de gestión del proceso selección e integración al empleo del capital humano en una agencia empleadora"*. Cuba : Contribuciones a la Economía, 2010. pág. 5. 5.
11. **Gérald Destinobles, A.** *El capital humano en las teorías del crecimiento económico*. Cuba : Edición electrónica gratuita, 2006. pág. 4. 10.
12. **Albert Sunne Torrents, Francisco Gil Vilda, Ignasi Arcusa Postils.** *Manual Practico de Disenno de Sistemas Productivos*. Cuba : Ediciones Diaz de Santos, 2004. 8479786426.
13. **Acuña, Albert.** MercadoJobs. *MercadoJobs*. [En línea] BECITHEW SA DE CV. [Citado el: 12 de Enero de 2015.] <http://cu.mercadojobs.com/>.
14. **Miguel Obregon, Albert Acuña.** Computrabajo. *Computrabajo*. [En línea] Edimburgo, Escocia, 23 de Mayo de 2000. [Citado el: 24 de Enero de 2015.] <http://www.computrabajo.com/>.
15. **George Sebastian Coll, Elena Jeferson Estolklin.** Bumeran. *Bumeran*. [En línea] 12 de Abril de 2001. [Citado el: 23 de Enero de 2015.] <http://www.bumeran.com/postulantes/bifurcador.bum>.

16. **Ing Jonatan Stallon, Ing Jefree Andersen.** Acciontrabajo. *Acciontrabajo*. [En línea] W3 Ltd, 5 de Noviembre de 2001. [Citado el: 24 de Enero de 2015.] <http://cu.acciontrabajo.com/>.
17. **Lugones, Elexey Hernández.** Intellego. *Intellego*. [En línea] IDC México, 10 de Marzo de 2000. [Citado el: 11 de Marzo de 2015.] <http://www.intellego.com.mx/es/erp-planificacion-de-recursos-empresariales..>
18. **Lic. Miguel P. Cabrera González, Guillermo Obregon Rodriguez.** *XV Forum de Ciencia y Técnica Sistema Economico Integrado Verast Sarasola*. Villa Clara : Casa de Desoft Villa Clara, 2000.
19. **Gonzalez, Gilberto.** Rodas XXI. *Rodas XXI*. [En línea] CIMATEL, 20 de Enero de 2002. [Citado el: 3 de Marzo de 2015.] <http://www.rodasxxi.cu/>.
20. **Lugonez, Elexey Hernández.** Monografias. *Monografias*. [En línea] UCI, 15 de Septiembre de 2009. [Citado el: 18 de Febrero de 2015.] <http://www.monografias.com/trabajos95/sistemas-erp-y-su-implicacion-cuba/sistemas-erp-y-su-implicacion-cuba.shtml..>
21. **Romer, Hermes.** *Metodologias de Desarrollo*. España : Administraciones publicas, 2006.
22. **Acuña, Kareenny Brito.** Eumed. *Eumed*. [En línea] Metodologias de Desarrollo,. [Citado el: 28 de Enero de 2015.] <http://www.eumed.net/libros/2009c/584/RUP%20Diseno%20e%20implementacion%20del%20sistema.htm>.
23. **Calero, Solís. Manuel.** WillyDev. *WillyDev*. [En línea] 10 de Mayo de 2008. [Citado el: 10 de Febrero de 2015.] <http://www.willydev.net/descargas/prev/ExplicaXp.pdf>.

24. **Pearson Addison-Wesley, Martin. R, Newkirk, J.** Universo. *Universo*. [En línea] 11 de Junio de 2012. [Citado el: 26 de Enero de 2015.] http://www.uv.mx/universo/486/infgral/infgral_15.html..
25. **Penadés, Patricio Letelier y M^a Carmen.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Valencia España : Universidad Politécnica de Valencia Camino de Vera, 2012. 46022.
26. **Asenjo, Jorge Sánchez.** Servidores de Aplicaciones Web. *Servidores de Aplicaciones Web*. [En línea] 11 de Febrero de 2011. [Citado el: 10 de Febrero de 2015.] <http://www.jorgesanchez.net>..
27. **Counte, Wilbert.** Ibrugor. *Ibrugor*. [En línea] 11 de Junio de 2014. [Citado el: 12 de Febrero de 2015.] <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>.
28. **Álvarez, Sara.** Desarrolloweb. *Desarrolloweb*. [En línea] 20 de Septiembre de 2010. [Citado el: 12 de Diciembre de 2014.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>..
29. **Guerrero, Rafael Martinez.** PostgreSQL-es. *PostgreSQL-es*. [En línea] 12 de Junio de 2011. [Citado el: 20 de Diciembre de 2014.] <http://www.postgresql.org/es/>..
30. **Fuster, G. G., J. M. F. Torres.** *Evaluación comparativa de herramientas CASE para UML desde el punto de vista notacional. Tecnología de Objetos Secciones Técnicas*. Madrid España : Dpto de Informática, Universidad Carlos III de Madrid, 2006.
31. **Graterol, Yohan.** CristalLab. *CristalLab*. [En línea] 10 de Marzo de 2011. [Citado el: 15 de Diciembre de 2014.] <http://www.cristalab.com/tutoriales/pycharm-el-mejor-ide-para-tus-proyectos-en-python-c114084/>..

32. **Alonso, Evelyn Menendez.** *Introducción a Herramientas CASE*. Valencia. España : EUI-FI. Universidad Politécnica de Valencia , 2011.
33. **Freedown Load Manage.** *Freedown Load Manage*. [En línea] [Citado el: 18 de Diciembre de 2014.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/)
34. **Almagro, Carlos Ureña.** *Lenguajes de Programacion*. España : Universidad de España, 2012.
35. **Martelli, Bruno Gorjón Salvador y Alex.** *Python. Guía de referencia*. España : Anaya Multimedia-Anaya Interactiva, 2007. 978-84-415-2317-3.
36. **Gutierrez, Javier J.** *Que es un Framework*. España : s.n., 2005.
37. **Newman, Scott.** *Django 1.0 Template Development*. 2008.
38. **Newman, Scott.** *Django 1.0 Template Development*. 2008. pág. 4. 7.
39. **Mora, Miguel A.** Slidshare. *Slidshare*. [En línea] 24 de Marzo de 2008. [Citado el: 10 de Febrero de 2015.] <http://es.slideshare.net/MiquelMora/historias-de-usuario>.
40. **Casanova, Josep.** Desarrolloweb. *Desarrolloweb*. [En línea] 9 de Septiembre de 2004. [Citado el: 28 de Enero de 2015.] <http://www.desarrolloweb.com/articulos/1622.php>.
41. **Acuña, Sebastan.** Solinfo. *Solinfo*. [En línea] Ecuador, 22 de Marzo de 2000. [Citado el: 10 de Marzo de 2015.] http://www.solinfo.cl/wp-content/themes/zine-style/archivos/cliente_servidor.html.

- 42. Welicki, Leon.** msdn. *msdn*. [En línea] EUA, 7 de Marzo de 2000. [Citado el: 12 de Marzo de 2015.] <http://msdn.microsoft.com/es-es/library/bb972251.aspx..>
- 43. Moreno, Sergio Infante.** Maestros del Web. *Maestros del Web*. [En línea] 30 de Abril de 2012. [Citado el: 12 de Marzo de 2015.] [http://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django/..](http://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django/)
- 44. Arangueren, Gilber.** Ingenierossoftware. *Ingenierossoftware*. [En línea] 20 de Agosto de 2003. [Citado el: 23 de Marzo de 2015.] <http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php..>
- 45. Dr. José Narro Robles, Dr. Felipe Bracho Carpizo.** Recursos Web. *Recursos Web*. [En línea] Dirección General de Cómputo y de Tecnologías de Información y Comunicación. , 23 de Febrero de 2003. [Citado el: 5 de Mayo de 2015.] [http://recursosweb.unam.mx/recursos-web/creacion-de-paginas-web/estandares-de-codificacion/.](http://recursosweb.unam.mx/recursos-web/creacion-de-paginas-web/estandares-de-codificacion/)
- 46. Sommariva, Andres.** MicroGestion. *MicroGestion*. [En línea] 24 de Febrero de 2014. [Citado el: 12 de Mayo de 2015.] <http://www.microgestion.com/index.php/mg-developers/articulos/74-unit-test-part1-mock>.
- 47. J. J. Gutiérrez, M. J. Escalona, M. Mejías, J. Torres.** *Prueba del Sistema en Programación Extrema*. Sevilla España : Departamento de Lenguajes y Sistemas Informaticos. Universidad de Sevilla, 2006.
- 48. Sara Cortina, Francisco Alarcon.** Diclib. *Diclib*. [En línea] España, 10 de Abril de 2008. [Citado el: 20 de Marzo de 2015.] http://www.diclib.com/cgi-bin/d1.cgi?l=es&base=es_wiki_10&page=showid&id=53629..

Bibliografía

1. **Jardinez, Garcia Dousat y Reyes.** *"Cuadro de mando integral para el control de gestión del proceso selección e integración al empleo del capital humano en una agencia empleadora"*. Cuba : Contribuciones a la Economía, 2010. pág. 2.
2. **Gérald Destinobles, A.** *El capital humano en las teorías del crecimiento económico*. Cuba : Edición electrónica gratuita, 2006.
3. **Garcia Dousat, Reyes Jardinez y Díaz Gamboa.** *"¿Cómo diagnosticar el Sistema de Gestión Integrada de Capital Humano?"*. Cuba : Observatorio de la Economía Latinoamericana, 2011. 148.
4. **Jardinez, Garcia Dousat y Reyes.** *"Cuadro de mando integral para el control de gestión del proceso selección e integración al empleo del capital humano en una agencia empleadora"*. Cuba : Contribuciones a la Economía, 2010. pág. 3.
5. **Reyes, Raul.** *Empleare. Empleare*. [En línea] Valencia, 4 de Noviembre de 2009. [Citado el: 5 de Diciembre de 2014.] <http://www.empleare.com/que-es-un-contrato-laboral.html>.
6. **E., Fernando Vidal.** *Guía para la Contratación*. Andalucía España : Consejería de Medio Ambiente, IDEAS, Bakeaz, 2006.
7. **Antunes, Rafael Obrego.** *Comrapublicaetica. Comrapublicaetica*. [En línea] Villafranca de Córdoba, ciudad Cordoba España, 13 de Diciembre de 2002. [Citado el: 19 de Diciembre de 2014.] <http://www.comrapublicaetica.org/index.php/le-ayudamos-paso-a-paso/proceso-de-contratacion..>

8. **Jardínez, Garcia Dousat y Reyes.** *"Cuadro de mando integral para el control de gestión del proceso selección e integración al empleo del capital humano en una agencia empleadora"*. Cuba : Contribuciones a la Economía, 2010. pág. 4. 5.
9. **Jardínez., MSc. Ing. Yordanis Garcia Dousat. Ing. Lixania Reyes.** *PROCEDIMIENTO PARA LA GESTIÓN DE SELECCIÓN E INTEGRACIÓN AL EMPLEO DEL CAPITAL HUMANO DESDE LA PERSPECTIVA DE UNA AGENCIA EMPLEADORA*. Valencia : Bekaez, 2006.
10. **Jardínez, Garcia Dousat y Reyes.** *"Cuadro de mando integral para el control de gestión del proceso selección e integración al empleo del capital humano en una agencia empleadora"*. Cuba : Contribuciones a la Economía, 2010. pág. 5. 5.
11. **Gérald Destinobles, A.** *El capital humano en las teorías del crecimiento económico*. Cuba : Edición electrónica gratuita, 2006. pág. 4. 10.
12. **Albert Sunne Torrents, Francisco Gil Vilda, Ignasi Arcusa Postils.** *Manual Practico de Disenno de Sistemas Productivos*. Cuba : Ediciones Diaz de Santos, 2004. 8479786426.
13. **Acuña, Albert.** MercadoJobs. *MercadoJobs*. [En línea] BECITHEW SA DE CV. [Citado el: 12 de Enero de 2015.] <http://cu.mercadojobs.com/>.
14. **Miguel Obregon, Albert Acuña.** Computrabajo. *Computrabajo*. [En línea] Edimburgo, Escocia, 23 de Mayo de 2000. [Citado el: 24 de Enero de 2015.] <http://www.computrabajo.com/>.
15. **George Sebastian Coll, Elena Jeferson Estolklin.** Bumeran. *Bumeran*. [En línea] 12 de Abril de 2001. [Citado el: 23 de Enero de 2015.] <http://www.bumeran.com/postulantes/bifurcador.bum>.

16. **Ing Jonatan Stallon, Ing Jefree Andersen.** Acciontrabajo. *Acciontrabajo*. [En línea] W3 Ltd, 5 de Noviembre de 2001. [Citado el: 24 de Enero de 2015.] <http://cu.acciontrabajo.com/>.
17. **Lugones, Elexey Hernández.** Intellego. *Intellego*. [En línea] IDC México, 10 de Marzo de 2000. [Citado el: 11 de Marzo de 2015.] <http://www.intellego.com.mx/es/erp-planificacion-de-recursos-empresariales..>
18. **Lic. Miguel P. Cabrera González, Guillermo Obregon Rodriguez.** *XV Forum de Ciencia y Técnica Sistema Economico Integrado Verast Sarasola*. Villa Clara : Casa de Desoft Villa Clara, 2000.
19. **Gonzalez, Gilberto.** Rodas XXI. *Rodas XXI*. [En línea] CIMATEL, 20 de Enero de 2002. [Citado el: 3 de Marzo de 2015.] <http://www.rodasxxi.cu/>.
20. **Lugonez, Elexey Hernández.** Monografias. *Monografias*. [En línea] UCI, 15 de Septiembre de 2009. [Citado el: 18 de Febrero de 2015.] <http://www.monografias.com/trabajos95/sistemas-erp-y-su-implicacion-cuba/sistemas-erp-y-su-implicacion-cuba.shtml..>
21. **Romer, Hermes.** *Metodologias de Desarrollo*. España : Administraciones publicas, 2006.
22. **Acuña, Kareenny Brito.** Eumed. *Eumed*. [En línea] Metodologias de Desarrollo,. [Citado el: 28 de Enero de 2015.] <http://www.eumed.net/libros/2009c/584/RUP%20Diseno%20e%20implementacion%20del%20sistema.htm>.
23. **Calero, Solís. Manuel.** WillyDev. *WillyDev*. [En línea] 10 de Mayo de 2008. [Citado el: 10 de Febrero de 2015.] <http://www.willydev.net/descargas/prev/ExplicaXp.pdf>.

24. **Pearson Addison-Wesley, Martin. R, Newkirk, J.** Universo. *Universo*. [En línea] 11 de Junio de 2012. [Citado el: 26 de Enero de 2015.] http://www.uv.mx/universo/486/infgral/infgral_15.html..
25. **Penadés, Patricio Letelier y M^a Carmen.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Valencia España : Universidad Politécnica de Valencia Camino de Vera, 2012. 46022.
26. **Asenjo, Jorge Sánchez.** Servidores de Aplicaciones Web. *Servidores de Aplicaciones Web*. [En línea] 11 de Febrero de 2011. [Citado el: 10 de Febrero de 2015.] <http://www.jorgesanchez.net>..
27. **Counte, Wilbert.** Ibrugor. *Ibrugor*. [En línea] 11 de Junio de 2014. [Citado el: 12 de Febrero de 2015.] <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>.
28. **Álvarez, Sara.** Desarrolloweb. *Desarrolloweb*. [En línea] 20 de Septiembre de 2010. [Citado el: 12 de Diciembre de 2014.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>..
29. **Guerrero, Rafael Martinez.** PostgreSQL-es. *PostgreSQL-es*. [En línea] 12 de Junio de 2011. [Citado el: 20 de Diciembre de 2014.] <http://www.postgresql.org/es/>..
30. **Fuster, G. G., J. M. F. Torres.** *Evaluación comparativa de herramientas CASE para UML desde el punto de vista notacional. Tecnología de Objetos Secciones Técnicas*. Madrid España : Dpto de Informática, Universidad Carlos III de Madrid, 2006.
31. **Graterol, Yohan.** CristalLab. *CristalLab*. [En línea] 10 de Marzo de 2011. [Citado el: 15 de Diciembre de 2014.] <http://www.cristalab.com/tutoriales/pycharm-el-mejor-ide-para-tus-proyectos-en-python-c114084/>..

32. **Alonso, Evelyn Menendez.** *Introducción a Herramientas CASE*. Valencia. España : EUI-FI. Universidad Politécnica de Valencia , 2011.
33. **Freedown Load Manage.** *Freedown Load Manage*. [En línea] [Citado el: 18 de Diciembre de 2014.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/)
34. **Almagro, Carlos Ureña.** *Lenguajes de Programacion*. España : Universidad de España, 2012.
35. **Martelli, Bruno Gorjón Salvador y Alex.** *Python. Guía de referencia*. España : Anaya Multimedia-Anaya Interactiva, 2007. 978-84-415-2317-3.
36. **Gutierrez, Javier J.** *Que es un Framework*. España : s.n., 2005.
37. **Newman, Scott.** *Django 1.0 Template Development*. 2008.
38. **Newman, Scoot.** *Django 1.0 Template Development*. 2008. pág. 4. 7.
39. **Mora, Miguel A.** Slidshare. *Slidshare*. [En línea] 24 de Marzo de 2008. [Citado el: 10 de Febrero de 2015.] <http://es.slideshare.net/MiquelMora/historias-de-usuario>.
40. **Casanova, Josep.** Desarrolloweb. *Desarrolloweb*. [En línea] 9 de Septiembre de 2004. [Citado el: 28 de Enero de 2015.] <http://www.desarrolloweb.com/articulos/1622.php>.
41. **Acuña, Sebastan.** Solinfo. *Solinfo*. [En línea] Ecuador, 22 de Marzo de 2000. [Citado el: 10 de Marzo de 2015.] http://www.solinfo.cl/wp-content/themes/zine-style/archivos/cliente_servidor.html.

-
- 42. Welicki, Leon.** msdn. *msdn*. [En línea] EUA, 7 de Marzo de 2000. [Citado el: 12 de Marzo de 2015.] <http://msdn.microsoft.com/es-es/library/bb972251.aspx..>
- 43. Moreno, Sergio Infante.** Maestros del Web. *Maestros del Web*. [En línea] 30 de Abril de 2012. [Citado el: 12 de Marzo de 2015.] [http://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django/..](http://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django/)
- 44. Arangueren, Gilber.** Ingenierosoftware. *Ingenierosoftware*. [En línea] 20 de Agosto de 2003. [Citado el: 23 de Marzo de 2015.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php..>
- 45. Dr. José Narro Robles, Dr. Felipe Bracho Carpizo.** Recursos Web. *Recursos Web*. [En línea] Dirección General de Cómputo y de Tecnologías de Información y Comunicación. , 23 de Febrero de 2003. [Citado el: 5 de Mayo de 2015.] [http://recursosweb.unam.mx/recursos-web/creacion-de-paginas-web/estandares-de-codificacion/.](http://recursosweb.unam.mx/recursos-web/creacion-de-paginas-web/estandares-de-codificacion/)
- 46. Sommariva, Andres.** MicroGestion. *MicroGestion*. [En línea] 24 de Febrero de 2014. [Citado el: 12 de Mayo de 2015.] <http://www.microgestion.com/index.php/mg-developers/articulos/74-unit-test-part1-mock>.
- 47. J. J. Gutiérrez, M. J. Escalona, M. Mejías, J. Torres.** *Prueba del Sistema en Programación Extrema*. Sevilla España : Departamento de Lenguajes y Sistemas Informaticos. Universidad de Sevilla, 2006.
- 48. Sara Cortina, Francisco Alarcon.** Diclib. *Diclib*. [En línea] España, 10 de Abril de 2008. [Citado el: 20 de Marzo de 2015.] http://www.diclib.com/cgi-bin/d1.cgi?l=es&base=es_wiki_10&page=showid&id=53629..

Anexos

Anexo 1 Historias de Usuarios

Tabla 17 HU # 4: Completar los Datos Personales

Historia de usuario	
Número: 4	Usuario: Aspirante
Nombre de la Historia de Usuario: Completar los Datos Personales	
Prioridad en el negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1\3	Iteración asignada:
Programador responsable: Jorge Ameth García Diago Melquiades Megret Ramírez	
Descripción: Permite a los usuarios llenar los campos para poder crear una solicitud de una plaza determinada.	
Observaciones:	
Prototipo de interfaz:	

Tabla 18 HU # 5: Completar otros Datos

Historia de usuario	
Número: 5	Usuario:

Nombre de la Historia de Usuario: Completar otros Datos	
Prioridad en el negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1\3	Iteración asignada:
Programador responsable: Jorge Ameth García Diago Melquiades Megret Ramírez	
Descripción: Permite a los usuarios llenar otros datos para la creación de las solicitudes.	
Observaciones:	
Prototipo de interfaz:	

Tabla 19 HU # 6: Gestionar Solicitudes por Departamentos

Historia de usuario	
Número: 6	Usuario: Aspirante
Nombre de la Historia de Usuario: Gestionar Solicitudes por Departamentos	
Prioridad en el negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1\3	Iteración asignada:
Programador responsable: Jorge Ameth García Diago Melquiades Megret Ramírez	
Descripción: Permite a los usuarios gestionar la información en cuanto a sus solicitudes.	



Observaciones:
Prototipo de interfaz:

Tabla 20 HU # 7: Filtrar Ofertas por Departamentos

Historia de usuario	
Número: 7	Usuario: Aspirante
Nombre de la Historia de Usuario: Filtrar Ofertas por Departamentos	
Prioridad en el negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1\3	Iteración asignada:
Programador responsable: Jorge Ameth García Diago Melquiades Megret Ramírez	
Descripción: Permite a los usuarios ver las ofertas teniendo en cuenta un departamento determinado.	
Observaciones:	
Prototipo de interfaz:	

Tabla 21 HU # 8: Revisar Mensajes y Notificaciones

Historia de usuario	
Número: 8	Usuario: Aspirante

Nombre de la Historia de Usuario: Revisar Mensajes y Notificaciones	
Prioridad en el negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1\3	Iteración asignada:
Programador responsable: Jorge Ameth García Diago Melquiades Megret Ramírez	
Descripción: Permite a los usuarios poder tener conocimiento sobre si alguna de las solicitudes creadas por él han sido aprobadas.	
Observaciones:	
Prototipo de interfaz:	

Tabla 22 HU # 9: Realizar Contratación

Historia de usuario	
Número: 9	Usuario: Jefe de Departamento
Nombre de la Historia de Usuario: Realizar Contratación	
Prioridad en el negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 1\3	Iteración asignada: 2
Programador responsable: Jorge Ameth García Diago Melquiades Megret Ramírez	



Descripción: El Jefe de departamento después de haber seleccionado un aspirante procede a añadirlo como un nuevo trabajador del departamento.

Observaciones:

Prototipo de interfaz: