



Sistema de Gestión de Recursos Humanos
de la Unión de Empresas de Recuperación de Materias Primas.

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

Autores: Lilianne Reyes Vargas.

Diego Quesada Guerrero.

Tutora: Ing. Marianny Hernández Batista.

Ciudad de La Habana, junio 2010.

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas, para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2010.

Autores:

Lilianne Reyes Vargas.
Guerrero.

Diego Quesada

Firma

Firma

Tutor:

Ing. Marianny Hernández Batista.

Firma

Dedicatoria

A mis papás Manolo y Canta y a mis madres Esmeralda y Leguis.

A mi hermana Coqui, a mis tías Vilma y Chicha y a mis primos Yunion y Yaimara.

Lilianne.

A mi papá, mi mamá, mi hermana y mi novia Daro.

A mis abuelos.

Diego.

Agradecimientos

A mi mamá, por todos los años que se ha dedicado a mí, por apoyarme en todos mis sueños y por exigirme los mejores resultados en cada momento, gracias mami por todo tu amor, por tu dedicación día tras día, tú eres una de esas tantas personas que en mi vida representan algo especial, gracias mami por aguantar mis malacrianzas, que no han sido pocas, tú sabes que aunque fuerte la mayor parte del tiempo tu hijita siempre te querrá. A ti te agradeceré siempre y hoy estoy aquí para decirte lo logré mami ya soy Ingeniera.

A mi abuela Esmeralda, mi mamá más favorita de las tantas que tengo. Gracias mami por siempre estar ahí para mí, por tus consejos que nunca me han faltado aunque a veces reiterativos tal vez por lo cabecidura que soy, gracias mami por la enseñanza que has sabido transmitirme por tu apoyo en todo momento en los buenos y en los malos, por tenerme como tu nieta consentida y malcriada, gracias por el abundante amor que has sabido darme y espero nunca me falte porque sino me faltaría el pedazo más importante de mi vida, a ti mamita linda siempre te estaré agradecida.

A mi tía Vilma, más que tía mi tercera mamá, a ti mi tía del alma gracias por todo tu apoyo, por saberme entender cuando nadie lo hace, por tus consejos que siempre llegaron en el momento que más los necesité, gracias tía por todo el cariño que me has dado todos estos años, gracias porque a pesar de ser la nazi de la familia como dice Coquí me dejaste a mí la mejor parte donde puedo seguir siendo tu consentida, bueno compartiendo el gusto con la Patry que hasta Yuni se pone celoso aunque lo niegue. A ti mi tía preciosa y de oro gracias.

A Canta por todos los años que ha dedicado en criarme y por tenerme como una hija más. A mi hermana por quererme de la misma forma que la quiero a ella. A mi tía Chicha por aceptarme todos mis antojos. A mis primos en especial a Yuni y Maidolis por todo el cariño que me han dado. A Yaimara, Tomy, Julio, Fernando y Emet por quererme y sentirse orgullosos de mi.

A mi papá y mis abuelos por todo el cariño que me han dado a pesar de la distancia. A mi hermano por siempre confiar en mí.

Agradecimientos

Quiero agradecer en primer lugar a mi mamá y mi papá por apoyarme en todo momento y saber guiarme por el camino correcto. A mi hermana por cuidarme, quererme mucho y darme un sobrino bello. A todos aquellos familiares que ya no están presentes y que quiero mucho. A mis abuelas que las quiero con la vida y a toda mi familia.

Un agradecimiento muy especial a mi novia Daro que me ha acompañado todo este tiempo y a sus padres que me han acogido como al hijo varón que no tuvieron.

A mis hermanos de universidad Dariel, Alain y Javier.

A mis hermanas de universidad Daily y Elizabeta.

A mi compañera de tesis por soportarme.

Resumen

En el departamento de recursos humanos de la Unión de Empresas de Recuperación de Materias Primas, existen dificultades con la manipulación de documentos debido a la gran cantidad de información que allí se guarda. A raíz de este impedimento, existe la duplicación y pérdida de los archivos y se cometen errores en su procesamiento.

En Cuba existen variedades de *software* que son comercializados por diferentes organismos, pero al igual que aquellos que existen a nivel mundial, son vendidos por grandes sumas de dinero o en ocasiones no satisfacen las necesidades del cliente.

El presente trabajo surge con la necesidad de mitigar el tiempo que se gasta innecesariamente, en la labor con los documentos referentes al capital humano de forma manual. Comprende la elaboración de una herramienta capaz de agilizar los procesos de reclutamiento y selección en la entidad. El usuario a través del sistema podrá gestionar las solicitudes de fuerza laboral, las convocatorias, los candidatos interesados en una solicitud y los trabajadores incorporados a la entidad.

El producto será capaz de cumplir con la necesidad de su fabricación y permitirá a sus consumidores trabajar en un ambiente más cómodo, sin que exista la posibilidad de procesar un error en su información.

Este documento recoge el resultado de todo el trabajo realizado para la elaboración del sistema antes mencionado, incluyendo el estudio del estado del arte de aplicaciones con objetivos similares, el estudio y definición de las características del sistema, así como la planificación, implementación, prueba y la factibilidad del producto.

Introducción	1
Capítulo 1: Fundamentación Teórica	6
Introducción	6
1.1 Recursos Humanos (RRHH).....	6
1.2 Gestión de RRHH.....	6
1.3 Software de gestión.....	6
1.4 Software de gestión de recursos humanos.....	7
1.5 Ejemplo de software de gestión de recursos humanos en el mundo.....	7
1.6 Ejemplo de software de gestión de recursos humanos en Cuba.....	9
1.7 Tendencias, tecnologías y herramientas.....	11
1.7.1 Metodología de desarrollo.....	11
1.7.2 Arquitectura.....	13
1.7.3 Lenguajes de programación para la web.....	16
1.7.4 Servidores Web.....	18
1.7.5 Navegadores Web.....	20
1.7.6 Gestores de Bases de Datos	20
1.7.7 Frameworks (Marco de trabajo).....	21
1.7.8 Lenguaje de modelado.....	22
1.7.9 Herramientas de Desarrollo.....	22
1.7.10 Herramientas CASE.....	23
Conclusiones.....	26

Capítulo 2: Características del sistema. Fase de exploración y planificación	27
Introducción	27
2.1 Descripción de los procesos vinculados al campo de acción	27
2.2 Objeto de automatización	29
2.3 Propuesta del sistema	29
2.4 Personas relacionadas con el sistema	30
2.5 Requisitos Funcionales (RF)	31
2.6 Requisitos No Funcionales (RNF)	34
2.7 Fase de exploración	35
2.7.1 Historias de usuario (HU)	35
2.8 Fase de Planificación	41
2.8.1 Planificación de las Historias de Usuarios	41
2.8.2 Plan de Iteraciones	42
2.8.3 Plan de duración de las iteraciones	44
2.8.4 Plan de entregas	45
Conclusiones	45
Capítulo 3: Implementación y Pruebas	46
Introducción	46
3.1 Diseño del sistema	46
3.1.1 Tarjetas Clase, Responsabilidad y Colaboración (CRC)	46
3.1.2 Modelo de datos	51
3.2 Fase de Implementación	51

3.2.1 Tareas de Ingeniería	51
3.3 Diagrama de Despliegue.....	52
3.4 Fase de Pruebas	53
3.4.1 Pruebas de aceptación	54
Conclusiones.....	55
Capítulo 4: Estudio de Factibilidad.....	56
Introducción	56
4.1 Características del Proyecto	56
4.2 Cálculo de instrucciones fuertes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.....	59
4.2.1 Cálculo del esfuerzo nominal.....	59
4.2.2 Cálculo del esfuerzo ajustado.....	60
4.2.3 Cálculo del tiempo de desarrollo, cantidad de hombres y costo	62
4.3 Beneficios tangibles e intangibles	63
4.4 Análisis del costo	63
Conclusiones.....	63
Conclusiones generales	64
Recomendaciones	65
Bibliografía Referenciada	66
Bibliografía Consultada	70
Glosario de Términos.....	71

Tabla 2. 1 Personas relacionadas con el sistema.....	30
Tabla 2. 2 HU 1 Autenticar usuario	36
Tabla 2. 3 HU 2 Gestionar usuario	36
Tabla 2. 4 HU 3 Gestionar solicitud de fuerza laboral	37
Tabla 2. 5 HU 4 Gestionar convocatoria	37
Tabla 2. 6 HU 5 Gestionar candidato	37
Tabla 2. 7 HU 6 Gestionar trabajador	38
Tabla 2. 8 HU 7 Buscar	38
Tabla 2. 9 HU 8 Gestionar rol	39
Tabla 2. 10 HU 9 Gestionar área.....	39
Tabla 2. 11 HU 10 Gestionar cargo.....	40
Tabla 2. 12 HU 11 Gestionar grupo escala.....	40
Tabla 2. 13 HU 12 Gestionar categoría ocupacional	40
Tabla 2. 14 HU 13 Gestionar nivel escolar.....	41
Tabla 2. 15 HU 14 Mostrar permiso	41
Tabla 2. 16 Estimación de esfuerzos por historias de usuarios	42
Tabla 2. 17 Plan de duración de las iteraciones	45
Tabla 2. 18 Plan de duración de entregas	45
Tabla 3. 1 Tarjeta CRC clase <i>sfGuardAuth</i>	46
Tabla 3. 2 Tarjeta CRC clase <i>sfGuardUser</i>	47
Tabla 3. 3 Tarjeta CRC clase Solicitud	47
Tabla 3. 4 Tarjeta CRC clase Convocatoria.....	47
Tabla 3. 5 Tarjeta CRC clase Candidato.....	48
Tabla 3. 6 Tarjeta CRC clase Trabajador.....	48
Tabla 3. 7 Tarjeta CRC clase Buscar.....	49
Tabla 3. 8 Tarjeta CRC clase Área.....	49
Tabla 3. 9 Tarjeta CRC clase Cargo	49
Tabla 3. 10 Tarjeta CRC clase Rol.....	50
Tabla 3. 11 Tarjeta CRC clase Grupo_escala	50

Tabla 3. 12 Tarjeta CRC clase Categoría_ocupacional.....	50
Tabla 3. 13 Tarjeta CRC clase Nivel_escolar	50
Tabla 3. 14 Tarjeta CRC clase Permiso.....	51
Tabla 4. 1 Entradas externas.	57
Tabla 4. 2 Salidas externas.	57
Tabla 4. 3 Consultas externas.....	57
Tabla 4. 4 Archivos lógicos internos.....	58
Tabla 4. 5 Archivos de interfaz externos.	58
Tabla 4. 6 Puntos de Función desajustados.	59
Tabla 4. 7 Factor Escalar (Wi).....	60
Tabla 4. 8 Multiplicadores de esfuerzo (ME).....	61
Tabla 4. 9 Resultados.	62

Figura 1 Modelo Vista Controlador.....	25
Figura 2 Diagrama de despliegue.	53
Figura 3 Gráfico de prueba de aceptación.	55

Introducción

Hace dos décadas atrás se definía al término de recursos humanos (RRHH) como aquel que proporcionaba la fuerza laboral a una organización.

Actualmente, esa definición ha evolucionado y hasta ahora se considera a los RRHH como el estudio que ayuda a las organizaciones a obtener, desarrollar, mantener y conservar el número y el tipo adecuado de colaboradores. (1)

Otra definición moderna más concisa señala que la administración de los RRHH involucra todas las decisiones gerenciales y prácticas que afectan e influyen a las personas. (1)

Los RRHH ahora son considerados como una pieza clave en el desarrollo de las empresas, ya que permite la realización de las metas de éstas, elevando su papel a una posición estratégica. (1)

La gestión de RRHH de una empresa tiene como objetivo fundamental proveer a la empresa de personal suficiente tanto en calidad, como con la calidad requerida en el momento adecuado y con la motivación suficiente.

Se entiende por el término **gestionar** lo siguiente:

- ✓ Planificar: proceso de establecer objetivos con el fin de alcanzar determinados resultados. (Establecimientos de objetivos, elaboración de planes, etc.)
- ✓ Organizar: proceso de dividir el trabajo y de coordinar el logro de resultados que tienen un propósito común.
- ✓ Dirigir: proceso de conducir y coordinar esfuerzos laborales de las personas que integran una organización. Función mediante la cual se ponen en marcha las tareas programadas.
- ✓ Controlar: proceso de supervisar las actividades y resultados, comparándolos con los objetivos y tomando las acciones correctivas, si son necesarias. (2)

Por la dinámica con que evoluciona la Informática en el mundo actual y el carácter estratégico de la misma para el desarrollo económico-social de cualquier país, Cuba ha intensificado desde temprano la necesidad de introducir en la práctica social las nuevas Tecnologías de la Información y las Comunicaciones (TIC) y lograr una cultura informática lo que facilitaría a nuestra sociedad alcanzar el objetivo de un desarrollo sostenible.

En la actualidad las organizaciones buscan cada vez con mayor necesidad, formas de gestionar los RRHH a través de vías más efectivas con el fin de perfeccionar la gestión de los mismos con las exigencias del mundo de hoy. Esta operación se ha convertido en uno de los procesos que más atención acumula por parte de la dirección de las empresas.

Después del triunfo de la revolución, el Che siendo Ministro de Industrias crea la estrategia cubana basada en la recuperación de materias primas creando en aquel momento la llamada Empresa Consolidada de Recuperación de Materias Primas, la cual transitó históricamente por varios organismos centrales desde su fundación. En 1981 finalmente se convirtió en la Unión de Empresas de Recuperación de Materias Primas (UERMP), momento a partir del cual se fortalece la actividad económica y tecnológicamente del país.

Los métodos tradicionales manuales para mantener la información del personal no son viables en una organización relativamente grande. En la UERMP, los documentos que competen al departamento de RRHH se encuentran en archivos manuales y el volumen es considerable porque corresponde a documentos generales a través de los años de funcionamiento de la institución. Generar cualquier tipo de información se hace difícil, pues el proceso es manual e implica horas de trabajo, duplicación, errores y pérdidas de documentos por agotamiento del personal. La UERMP a través del departamento de RRHH, necesita gestionar fácilmente la información referente al personal, de tal forma responder con rapidez a las diversas consultas generando información confiable, precisa y oportuna. También es de fundamental importancia la consolidación de la información del personal administrativo y de servicio de la institución con el objetivo de facilitar el análisis y proporcionar elementos necesarios para una buena planificación.

El presente trabajo surge como necesidad de dar solución a la **situación problemática** antes expuesta, generando el siguiente **problema científico**:

¿Cómo mejorar la gestión de los Recursos Humanos en la Unión de Empresas de Recuperación de Materias Primas?

El **objeto de estudio** de la presente investigación es la gestión de recursos humanos en la Unión de Empresas de Recuperación de Materias Primas donde el **campo de acción** se define como la gestión de recursos humanos en la Unión de Empresas de Recuperación de Materias Primas para los procesos de reclutamiento y selección.

En términos generales, el **objetivo** es desarrollar un sistema informático que automatice la gestión de recursos humanos pertenecientes a la Unión de Empresas de Recuperación de Materias Primas del cual los **objetivos específicos** son:

- ✓ Realizar la fundamentación teórica.
- ✓ Automatizar los procesos de reclutamiento y selección.
- ✓ Obtener una aplicación *web* como solución factible a la situación problemática.
- ✓ Mantener la confiabilidad, integridad y disponibilidad del sistema.
- ✓ Brindar información que pueda utilizarse en procesos de auditorías.

Para guiar la investigación se plantea la siguiente **hipótesis**:

La implementación del Sistema de Gestión de recursos humanos de la Unión de Empresas de Recuperación de Materias Primas contribuirá a mejorar la gestión de los recursos humanos.

Para el cumplimiento de lo planteado se proponen las siguientes tareas científicas:

- ✓ Realizar el estudio del estado del arte de la investigación.
- ✓ Establecer las bases teóricas conceptuales de la investigación.
- ✓ Definir las tecnologías, herramientas, lenguajes y metodologías a utilizar.

- ✓ Realizar el análisis y diseño del sistema que se propone.
- ✓ Implementar el sistema.
- ✓ Realizar el estudio de factibilidad de la investigación

Para la ejecución de estas tareas científicas se utilizan los siguientes **métodos teóricos**:

- ✓ Análisis Histórico – Lógico: se estudia la evolución y desarrollo de la gestión de RRHH dentro de la empresa, además de estudiar los cambios que surgieron y dieron lugar a dicha situación.
- ✓ Analítico – Sintético: se hace una breve investigación y estudio detallado de documentos que se manejan en el proceso de gestión del personal en la empresa para así identificar el problema preciso existente en la misma, sus causas y lograr un profundo análisis y diseño.

Dentro de los **métodos empíricos** se emplean:

- ✓ Entrevista: se realiza una serie de entrevistas a las personas involucradas en el proceso de gestión de RRHH dentro de la UERMP, para tener un mejor conocimiento sobre los requisitos que debe tener el sistema.
- ✓ Observación: teniendo en cuenta los objetivos presentes en esta investigación pudo facilitar la evaluación centrada en los procesos de gestión, relacionados con dichos objetivos.
- ✓ Modelación: se realizaron modelos y diagramas con el objetivo de lograr una mejor visibilidad y entendimiento de cómo se debe desarrollar el sistema.

Este documento está estructurado en cuatro capítulos. A continuación se expone una breve descripción de los mismos:

Capítulo 1: Se aborda todo lo relacionado con la fundamentación teórica que sustenta la presente investigación acerca del estudio del arte, así como las tendencias, tecnologías y herramientas a utilizar para el desarrollo del sistema.

Capítulo 2: Se identifican y se describe el flujo de los procesos de reclutamiento y selección al igual que las características del sistema a través de los requisitos funcionales y no funcionales. También se describen las fases de exploración y planificación.

Capítulo 3: Se presentan las fases de implementación y pruebas propias de la metodología de desarrollo, además de los artefactos generados en cada fase.

Capítulo 4: Se realiza el estudio de factibilidad de la solución propuesta, calculándose el esfuerzo, tiempo de desarrollo y el costo del sistema.

Capítulo 1: Fundamentación Teórica

Introducción

En este capítulo se analizan algunos de los conceptos importantes para un mejor entendimiento. Se realiza una panorámica sobre las aplicaciones existentes en relación con el tema de la gestión de los recursos humanos donde se exponen sus principales características. También se hace un estudio de las herramientas y tecnologías candidatas para determinar cuáles de ellas son las más factibles para la realización de este trabajo y siguiendo los objetivos planteados.

1.1 Recursos Humanos (RRHH)

Cuando se utiliza el término “recurso humano” en singular, generalmente se refiere a las personas empleadas en una empresa u organización. Es sinónimo de “personal”. (3)

Sin embargo, cuando se emplea en plural “recursos humanos”, suele referirse al área de la administración que se ocupa de gerenciar al personal de una empresa. Esto incluye contratar, desarrollar, adiestrar, despedir, entre otras funciones. (3)

1.2 Gestión de RRHH

Es un proceso donde intervienen los miembros activos de la empresa, entendiéndose por tales: la dirección general con tareas de mando, los asalariados con la negociación de un contrato y los representantes del personal. (4)

1.3 Software de gestión

Las aplicaciones o *software* de gestión son aquellas diseñadas para sustituir uno o varios procedimientos, tanto comerciales como administrativos, que habitualmente realiza una persona en una empresa o institución de forma presencial, por un *software*, que permita realizar al cliente los mismos procedimientos de forma no presencial o disminuir el esfuerzo empleado para los mismos. (5)

El proceso de la información comercial constituye la mayor de las áreas de aplicación de *software* de gestión. Los sistemas discretos (por ejemplo: nóminas, cuentas de haberes-

débitos, inventarios, etc.), han evolucionado hacia el *software* de gestión, que accede a una o más bases de datos que contienen información comercial. Las aplicaciones en esta área reestructuran los datos existentes para facilitar las operaciones comerciales o gestionar la toma de decisiones. Además de las tareas convencionales de procesamiento de datos, las aplicaciones de *software* de gestión también realizan cálculo interactivo. (5)

1.4 Software de gestión de recursos humanos

Los *software* de gestión de recursos humanos están orientados a satisfacer diferentes necesidades de las empresas con el fin de gestionar al potencial humano dentro de las mismas, sus principales y más comunes funcionalidades son la planificación, el pago a los trabajadores, selección del personal, detectar las necesidades de capacitación, administrar los cursos de capacitación de la empresa, etc.

1.5 Ejemplo de software de gestión de recursos humanos en el mundo

RRHH: es una línea de productos de alquiler que se contrata en forma anual y permite a la empresa recibir currículos de postulantes a distintos empleos por *internet* de una manera ordenada y segura. Permite además capturas de currículos vitae (CV) por email con su formato estándar. (6)

Con RRHH se puede:

- ✓ Brindar información procesada en forma totalmente sistematizada y estandarizada.
- ✓ Evita la duplicación de currículos, es decir, tener de cada candidato más de un CV.
- ✓ Ahorra tiempo y horas hombre en la clasificación manual de currículos.
- ✓ Los datos son cargados en un formato estándar, es decir, el postulante vuelca sólo la información que la empresa requiere, evitando datos irrelevantes que ocupan espacios innecesarios. (6)

Este producto tiene como desventajas que es sólo para gestionar los CV dentro de una empresa no incluye el resto de los procesos que abarca el área de RRHH.

HumaNex: es una herramienta diseñada para ayudar y contribuir a la mejora de la gestión del proceso completo de selección y la gestión de ofertas. La plataforma *software* automatiza y acelera la mayor parte de los procesos asociados con el reclutamiento, desde la especificación y fijación de las vacantes, la gestión de CV y organización de cada etapa de la entrevista y del proceso de selección. (7)

HumaNex permite:

- ✓ Una gestión sencilla e integrada del proceso de selección.
- ✓ Ofrece potentes funcionalidades de búsqueda para la identificación de los candidatos más apropiados.
- ✓ Ofrece un completo informe de seguimiento de la acción de selección online. (7)

Esta herramienta tiene la característica de que hay que pagar una cuota de dinero al mes para su utilización, condición que la empresa no está dispuesta a cumplir.

Epsilon: es un producto realizado por la empresa española Grupo Castilla, especializada en el desarrollo de soluciones de *software*. Integra soluciones de gestión de nóminas y RRHH, portales corporativos del empleado y del candidato y cuadros de mando para las áreas de RRHH. Epsilon se enfoca a empresas de tamaño mediano y grande (más de 100 empleados), además de Sector Público, ya que proporciona soluciones que se ajustan perfectamente a las necesidades de estas entidades. (8)

Su fabricante mantiene acuerdos con compañías como *Microsoft*, *IBM* y *Oracle* para el desarrollo y puesta en marcha no sólo de Epsilon sino de diversos proyectos tecnológicos desarrollados por esta empresa, los mismos son firmados bajo licencias propietarias. (8)

IntegRHa: es una herramienta implementada por la compañía farmacéutica Almirall radicada en Barcelona. Con este *software* se ha conseguido ejecutar los distintos procesos de recursos humanos de la compañía en una plataforma única. Esta reside en la *intranet* corporativa, en el portal de los empleados. Desde ahí el propio empleado puede gestionar los procesos que le afectan: solicitar vacaciones, permisos, consulta de nominas, modificaciones de datos.

Asimismo, permite que haya una visión integrada por parte del gestor de RRHH y del *manager* de cara a sus empleados: gestión de permisos, vacaciones, selección, evaluación del desempeño. (9)

IntegRHa tiene como característica que sólo está diseñado para los empleados de esta compañía Almirall, dado que los clientes que esta compañía tiene en los diferentes países aún no tienen un acceso a este *software* ya que “En cada país tienen una solución diferente, que puede ser compatible o no con esta solución, así que, por el momento no hay una interfaz entre la herramienta local y la nueva plataforma”. (9)

1.6 Ejemplo de software de gestión de recursos humanos en Cuba

Sistema Automatizado de RRHH de CIMEX: es una aplicación *web* que se encuentra en su versión 2.2.4 y brinda servicios a todas las unidades de la empresa corporativa CIMEX. Este sistema radica en la *intranet* de la misma y al sistema tienen acceso sólo los responsables de RRHH en cada unidad a diferencia de otros sistemas que pueden acceder todos los usuarios de la empresa. Está estructurado de forma tal que a nivel central se pueda chequear el trabajo en las unidades.

Dentro de los módulos que ofrece están:

- ✓ Diseño Organizacional
- ✓ Reclutamiento y Selección
- ✓ Gestión Integral del Personal
- ✓ Gestión Integral de Remuneraciones e Incentivos.
- ✓ Contabilidad
- ✓ Seguridad del Trabajo
- ✓ Administración
- ✓ Reportes

Tiene como característica que es un sistema cuya interfaz es solamente visible dentro de esta empresa, y su solución es bastante puntual, adaptada a las necesidades de la misma.

Assets: es un sistema informático encargado de la gestión integral. Comercializado por la firma panameña D´MARCO S.A. y distribuido en Cuba por INFOMASTER, entidad informática perteneciente a la Empresa Nacional de Producción y Servicios a la Educación Superior del MES. Cuenta con varios módulos entre los cuales se encuentra el de Recursos Humanos que está concebido para calcular las nóminas y controlar los recursos laborales de una entidad. Desde este módulo se puede controlar íntegramente los recursos laborales: empleados, estructura organizativa de la entidad y plantilla. Siempre que se introducen altas, bajas y otros movimientos de los empleados, se actualiza automáticamente el registro de empleados y se genera los reportes correspondientes. Es posible modificar plantillas, introducir cambios en la estructura organizativa, crear nuevos cargos y realizar conversiones de plazas. (10)

Actualmente Assets cuenta con un total de 497 clientes dispersos en Cuba de los cuales uno de ellos es la UCI. (11). También tiene como desventaja que para su explotación es necesario comprar su licencia.

RH-Expert: es una herramienta comercializada por la casa de desarrollo de *software* HavaSoft de la Empresa de las Tecnologías de la Información y el Conocimiento (TEICO). El sistema está diseñado en base a una estructura jerárquica de entidades que permite el envío de la información de los Recursos Humanos. Las entidades subordinadas envían su información a la entidad superior donde se almacena, procesa y consolida. (12)

Entre los aportes que brinda el sistema se encuentran:

- ✓ Poseer un control centralizado de la información de los Recursos Humanos.
- ✓ Logra la disminución del tiempo de procesamiento a la hora de realizar un consolidado de la información en la empresa superior.
- ✓ Se podrá contar con un mayor volumen de información.
- ✓ Brinda sugerencias a la hora de tomar decisiones. (12)

RH-Expert tiene como desventaja que no puede trabajar en Linux trabaja solamente en *Windows* y es necesario adquirir una licencia para usar el *software*.

Luego de estudiar estos productos y teniendo en cuenta que la UERMP exige un sistema bastante puntual y adaptado a las necesidades vigentes en la misma, además de no tener que pagar ninguna licencia para su utilización y tomando en cuenta también la posibilidad de emigración de nuestro país al *software* libre, se observan las herramientas con las que son desarrollados estos productos y se concluye que todos quedan descartados como solución al problema a resolver, pero que pueden servir como base para la realización del *software* que se pretende desarrollar brindando algunas características que se pueden tomar en cuenta:

- ✓ Evitar la duplicación de datos.
- ✓ Brindar información procesada en forma totalmente sistematizada y estandarizada.
- ✓ Ahorrar tiempo y horas hombre en la clasificación manual de la información.
- ✓ Una gestión sencilla e integrada del proceso de selección.
- ✓ Se podrá contar con un mayor volumen de información.

1.7 Tendencias, tecnologías y herramientas

Para la realización de este trabajo se ha realizado un estudio de las tendencias y tecnologías actuales en el mundo que pudiesen ser candidatas para el desarrollo del sistema a implementar. Se ha realizado también una buena selección de cuáles metodologías y herramientas, sistemas de gestión de bases de datos y arquitectura, utilizar para el desarrollo del sistema.

1.7.1 Metodología de desarrollo

Una metodología de desarrollo es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo *software*.

RUP (Proceso Unificado Racional en español): es una metodología de desarrollo de *software* que permite seleccionar fácilmente el conjunto de componentes de proceso que se ajustan a

las necesidades específicas del proyecto. Se podrán alcanzar resultados predecibles unificando con procesos comunes que optimicen la comunicación y creen un entendimiento común para todas las tareas, responsabilidades y artefactos. Desde un único sitio *web* centralizado de intercambio, las plataformas, herramientas y expertos de dominios proveen los componentes del proceso necesario para el éxito. (13)

Junto con el Lenguaje Unificado de Modelado *UML*, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. En realidad no es un proceso con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. A continuación se exponen algunas de las principales características de esta metodología.

- ✓ Guiado por casos de uso: Los casos de uso son el instrumento para describir el comportamiento del *software* y extraer los casos de prueba con los que se valida el sistema.
- ✓ Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño, describe la arquitectura del producto a desarrollar.
- ✓ Iterativo e incremental: Durante todo el proceso de desarrollo se producen versiones superiores.
- ✓ Utiliza el *UML* como lenguaje de representación visual.

Esta metodología se divide en 4 fases de desarrollo de *software* en las cuales son desarrolladas mediante el ciclo de iteraciones de varios flujos de trabajo.

Fases:

Inicio: El objetivo de esta etapa es determinar la visión del proyecto.

Elaboración: En esta etapa el objetivo es determinar la arquitectura.

Construcción: En esta etapa el objetivo es llegar a alcanzar una funcionalidad operativa.

Transición: El objetivo de esta etapa es llegar a obtener el *release* (liberación) del producto.

Por otra parte, si se decide aplicar RUP a un proyecto pequeño:

- ✓ Sería tedioso realizar todas las fases en un período corto de tiempo, por lo que no se podría aplicar mucho lo de la interactividad en cada fase.
- ✓ Es necesario incluir más personas en el equipo de desarrollo.
- ✓ En RUP se gasta demasiado tiempo para pasar a la fase de desarrollo, considerando que esta es una de las fases más importantes de un proyecto. (14)

XP (Programación Extrema): es una de las metodologías de desarrollo de *software* más exitosas en la actualidad utilizada para proyectos de corto plazo, de corto equipo y cuyo plazo de entrega es pequeño. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. (15)

Esta metodología se incluye dentro de las clasificadas como ágiles, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. (16)

XP es basado en historiales de uso que al igual que RUP, define los detalles técnicos sin meterse con los detalles de implementación. (16)

La comunicación con el cliente es fluida (a través de su representante) después de cada iteración el cliente recibe un pedazo de programa funcional, así el cliente está informado permanentemente y puede intervenir rápidamente si el desarrollo se aleja de sus necesidades. (16)

1.7.2 Arquitectura

La arquitectura de *software* o también llamada arquitectura lógica de un programa, es una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. Así, las arquitecturas más universales son:

Capítulo 1: Fundamentación Teórica

Cliente-servidor: es aquella arquitectura en la que confluyen una serie de aplicaciones basadas en dos categorías que cumplen funciones diferentes. En este modelo, el cliente envía un mensaje solicitando un determinado servicio a un servidor y este envía uno o varios mensajes con la respuesta.

Cliente-servidor se recomienda, en particular, para redes que requieran un alto grado de fiabilidad. Sus principales ventajas son:

- ✓ Sus recursos son centralizados debido a que el servidor es el centro de la red y puede administrar los recursos que son comunes a todos los usuarios.
- ✓ Seguridad mejorada ya que la cantidad de puntos de entrada que permite el acceso a los datos no es importante.
- ✓ Administración a nivel de servidor ya que los clientes no juegan un papel importante en este modelo, requiere menos administración.
- ✓ Gracias a esta arquitectura, es posible quitar o agregar clientes sin afectar el funcionamiento de la red y sin necesidad de realizar mayores modificaciones. (17)

Esta arquitectura presenta las siguientes desventajas:

- ✓ Cada servidor consta de una base de datos, por lo que es probable que una misma información se guarde en varios sitios, por lo que estará duplicada.
- ✓ Es fácil la congestión del tráfico cuando muchos clientes envían simultáneamente gran cantidad de peticiones al mismo servidor y puedes causarle problemas a este.
- ✓ Contiene pocas herramientas para la administración y ajuste del desempeño de los sistemas. (18)

Arquitectura de tres niveles: también llamada arquitectura de tres capas: presentación, negocio y datos, es una forma de programar en donde el objetivo es separar la lógica del negocio de la de diseño.

Entre sus ventajas se encuentran:

Capítulo 1: Fundamentación Teórica

- ✓ La interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC.
- ✓ El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en forma modular. Esto puede reducir los esfuerzos de desarrollo y mantenimiento, así como los costos de migración.
- ✓ La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.
- ✓ Separando la aplicación de la base de datos, hace más fácil utilizar nuevas tecnologías de agrupamiento y balance de cargas. (19)

Sus desventajas son:

- ✓ Pueden incrementar el tráfico en la red y requerir más balance de carga y tolerancia a las fallas.
- ✓ El costo aumenta porque requiere un servidor adicional para hospedar los componentes de desde donde se va a consumir toda la lógica de negocios, lo que trae consigo temas adicionales que se deben tener en cuenta como costo de licencias, costo de mantenimiento del servidor, etc.
- ✓ Se va a crear un brinco adicional cada vez que realice una llamada a los componentes de negocio que residen en este servidor de aplicaciones, lo que lleva a tener una respuesta más lenta. (20)

Arquitectura Orientada a Servicios (SOA): define la utilización de servicios para dar soporte a los requisitos del negocio y permite la creación de sistemas altamente escalables que reflejan el negocio de la organización.

Entre las ventajas de SOA se encuentra la reducción de tiempos y costos desde su desarrollo hasta su implantación y mantención, la flexibilidad, la escalabilidad, potencia las relaciones

con clientes y proveedores, la agilidad, la reducción de riesgos y la mayor calidad en los servicios. (21)

La implantación de SOA también tiene algunas desventajas, ya que requiere un cambio en las organizaciones, esto es, un alto esfuerzo, no siendo sencillo para la mayoría de las organizaciones. También requiere de algún de entrenamiento a nivel de diseño y de administración y por último, de no implementarse adecuadamente podría implicar inversiones significativas. (21)

Modelo Vista Controlador (MVC): describe una forma muy utilizada en la *web*, de organizar el código de una aplicación separando los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. (22)

Algunas de las ventajas de esta arquitectura son:

- ✓ Clara separación entre interfaz, lógica de negocio y presentación, que además provoca parte de las ventajas siguientes.
- ✓ Sencillez para crear distintas representaciones de los mismos datos.
- ✓ Facilidad para la realización de pruebas unitarias de los componentes, así como de aplicar desarrollo guiado por pruebas.
- ✓ Reutilización de los componentes.
- ✓ Simplicidad en el mantenimiento de los sistemas.
- ✓ Facilidad para desarrollar prototipos rápidos.
- ✓ Los desarrollados suelen ser más escalables. (23)

1.7.3 Lenguajes de programación para la web

Lenguaje del lado del cliente

Son aquellos que pueden ser directamente asimilados por el navegador y no necesitan un pre tratamiento.

HTML (Lenguaje de Marcado de Hipertexto): es el lenguaje usado por los navegadores para mostrar las páginas *web* al usuario, siendo hoy en día la interfaz más extendida en la red. Permite agrupar textos, sonidos e imágenes a gusto. (24)

Una de sus ventajas es que con respecto a los libros y revistas, permite introducción de referencias bibliográficas a otras páginas por medio de los enlaces hipertexto. (24)

Javascript: es un lenguaje de programación que puede ser utilizado por profesionales y para quienes inician en el desarrollo y diseño de sitios *web*. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. (25)

Tiene la ventaja de ser incorporado en cualquier página *web*, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado. (25)

CSS (Hoja de Estilo en Cascada): Las Hojas de Estilo en Cascada, permiten la creación de documentos visuales mucho más ricos que los que HTML nunca permitirá. No en vano, CSS está pensado única y exclusivamente para asistir al diseñador a la hora de dar estilo a un documento estructurado. (26)

La utilización de hojas de estilo CSS hace que el diseñador pueda reducir sustancialmente su carga de trabajo al diseñar todo un sitio. Esto se debe a que CSS es capaz de centralizar ciertos efectos visuales que se plasme en diversas secciones del sitio, en lugar de tenerlos diseminados por páginas y páginas del sitio. (26)

Lenguaje del lado del servidor

Son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él.

PHP: es un lenguaje de creación relativamente creciente que ha tenido gran aceptación dentro de la comunidad de *webmaster* (maestros del *web*), debido sobre todo a la potencia y simplicidad que lo caracteriza. (27)

Permite empapar sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados íntegramente en un lenguaje distinto al HTML. Ofrece un sinnúmero de funciones para la explotación de bases de datos de una manera llana, sin complicaciones. (27)

PHP, aunque es multiplataforma, ha sido concebido inicialmente para entornos *UNIX* y es en este sistema operativo donde se puede aprovechar mejor sus prestaciones. Es progresivamente construido por colaboradores desinteresados que implementan nuevas versiones del lenguaje. (27)

PHP permite al programador relacionarse con algunas de las bases de datos más importantes del mundo de la programación, tales como: *PostgreSQL*, *MySQL*.

ASP: es parecido a PHP en cuanto a dificultad. Realiza numerosas tareas sirviéndose de componentes que deben ser comprados por el servidor a determinadas empresas especializadas. Siendo una tecnología de *Microsoft*, está orientado hacia sistemas *Windows*, especialmente NT. (27)

Perl (Lenguaje Práctico para la Extracción e Informe): descrito por su creador Larry Wall como un excelente lenguaje optimizado para leer archivos de texto, extraer información de esos archivos y crear reportes basados en esa información, combinando lo mejor de C. (28)

Se considera un lenguaje interpretado ya que no es necesaria una previa compilación para poder ejecutarse, lo único que necesita es darle al intérprete, el código que se quiera que ejecute. (28)

1.7.4 Servidores Web

Un servidor *web* es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP (Protocolo de Transferencia de Hipertexto) o el protocolo HTTPS (versión cifrada y autenticada). Un servidor *web* básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente bucle:

1. Esperar peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80).
2. Recibe una petición.
3. Busca el recurso.
4. Envía el recurso utilizando la misma conexión por la que recibió la petición.
5. Vuele a segundo punto.

A partir del esquema anterior se han diseñado y desarrollado todos los servidores de HTTP que existen, variando sólo en el tipo de peticiones que pueden atender. (29)

Servidor HTTP Apache: es un servidor *web* HTTP de código abierto para plataformas *UNIX*, *Microsoft Windows*, *Macintosh* y otras. Presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por falta de una interfaz gráfica que ayude en su configuración. (30)

La arquitectura del *Apache* es muy modular. Multiplataforma, extensible, con código abierto y popular.

Servidor HTTP Cherokee: es un excelente servidor *web* gratuito, multiplataforma y de código abierto que se destaca por su velocidad, flexibilidad y facilidad de uso. (31)

Entre sus principales características se destacan:

- ✓ Es muy liviano
- ✓ Es seis veces más rápido que *Apache*
- ✓ Posee un panel de administración con una interfaz muy agradable.
- ✓ Es muy fácil de instalar y configurar.
- ✓ Está disponible bajo licencia GNU (Licencia Pública General) y puede ser instalado gratuitamente en cualquier computadora con sistema operativo *Linux* o *Windows*. (31)

1.7.5 Navegadores Web

Es un programa que permite visualizar la información que contiene una página *web* interpretando el código, HTML generalmente, en el que está escrita la página y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos.

Internet Explorer: conocido también como IE, es un navegador *web* desarrollado por *Microsoft* para el sistema operativo *Microsoft Windows*. Ha sido el navegador más utilizado desde 1999 pero ha caído paulatinamente debido a la competencia con otros navegadores principalmente *Mozilla Firefox*.

Mozilla: navegador multiplataforma que está disponible en varias versiones de sistemas operativos. Su código fuente es *software* libre. Superando a todas las versiones sumadas de *Internet Explorer*, es el navegador más utilizado en la actualidad. (32)

Incluye navegación por pestañas, corrector ortográfico, búsqueda progresiva, marcadores dinámicos, un administrador de descargas y un sistema de búsqueda integrado que utiliza el motor de búsqueda que desee el usuario. (32)

Tanto como personalización, seguridad, rendimiento, características y su misión, hacen que *Mozilla Firefox* sea el mejor navegador.

1.7.6 Gestores de Bases de Datos

Un Sistema de Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y un lenguaje de consulta. Un SGBD permite definir los datos de distintos niveles de abstracción y manipulación de dichos datos, garantizando la seguridad e integridad de los mismos.

MySQL: Es un sistema de gestión de bases de datos relacional utilizado en aplicaciones *web* y por herramientas de seguimiento de errores. Su popularidad como aplicación *web* está ligada a PHP. Es una base de datos rápida en la lectura cuando utiliza motor no transaccional, pero puede provocar problemas de integridad en entornos de alta concurrencia en la

modificación. En aplicaciones *web* hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lecturas de datos. (33)

PostgreSQL: Es un sistema de gestión de bases de datos relacional orientada a objetos y libre, publicado bajo una licencia de *software* libre permisiva. Como muchos otros proyectos de código abierto, el desarrollo de *PostgreSQL* no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. (34)

Está considerado como la base de datos de código abierto más avanzada del mundo. Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como *DB2* u *Oracle*. Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial). Tiene mejor soporte para *triggers* (disparador) y procedimientos en el servidor. Además, tiene ciertas características orientadas a objetos. (35)

1.7.7 Frameworks (Marco de trabajo)

En el desarrollo de *software*, un *framework* es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos de *software* concretos, mediante la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente, puede incluir soportes de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. (36)

Representa una arquitectura de *software* que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. (36)

Zend Framework: Se trata de un *framework* para desarrollo de aplicaciones *web* y servicios *web* con PHP, brinda soluciones para construir sitios *web* modernos, robustos y seguros. Es desarrollado por *Zend* que es la empresa que respalda comercialmente a PHP. (37)

Algunas de sus características son:

- ✓ Trabaja con MVC.
- ✓ Cuenta con módulos para manejar archivos *PDF* (Formato de Documento Portátil), canales *RSS* (Sindicación Realmente Simple), *Web Service* (Servicio Web), etc.
- ✓ El marco de *Zend* también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar, sin tener que escribir ninguna consulta *SQL*. (37)

Symfony: Es un *framework* para desarrollos basados en PHP5, totalmente *open source* (Código abierto), compatible con la mayoría de los ambientes, fácil de implementar y mantener y además implementa algunas de las tecnologías que más están en uso en estos días. (38)

1.7.8 Lenguaje de modelado

BPMN: la Notación de Modelado de Procesos de Negocio, es un nuevo estándar de modelado de procesos de negocio, donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación es diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes. (39)

UML (Lenguaje Unificado de Modelado): es una de las herramientas más emocionantes en el mundo actual del desarrollo de sistemas. Esto se debe a que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas. (40)

1.7.9 Herramientas de Desarrollo

Eclipse: Es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama “Aplicaciones de Cliente Enriquecido” opuesto a las aplicaciones “Cliente-liviano” basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar Entornos de Desarrollo Integrados (en inglés IDE). (41)

Es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado *Eclipse Modeling Project* (Proyecto

Modelado en Eclipse), cubriendo casi todas las áreas de *Model Driven Engineering* (Modelo de Ingeniería Impulsada). (41)

Netbeans PHP: herramienta de desarrollo que permite la creación y soporte de *script* (guión) PHP. El proyecto PHP es configurado para proporcionar el ambiente que se necesita. (42)

Algunas de sus características son:

- ✓ La mayor ventaja que ofrece es la edición mientras se codifica.
- ✓ Se crea una sincronización con el proyecto y el contenedor *web*.
- ✓ Permite la ejecución de procesos PHP. (42)

1.7.10 Herramientas CASE

Son un conjunto de métodos, utilidades que dan asistencia a los analistas, ingenieros de *software* y desarrolladores, durante todos los pasos de ciclo de vida de desarrollo de un *software* y facilitan el mejoramiento de ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. CASE es una sigla que corresponde a las iniciales de *Computer Aided Software Engineering* (*Ingeniería de Software Asistida por Ordenador*).

Rational Rose: es una herramienta de producción y comercialización establecida por *Rational Software Corporation* (Corporación Racional de *Software*), actualmente parte de *IBM* (Negocio Internacional de Máquinas). Rose es un instrumento operativo conjunto que utiliza el UML como medio para facilitar la captura de dominio de la semántica, la arquitectura y el diseño. Este *software* tiene la capacidad de crear, ver, modificar y manipular los componentes de un modelo. (43)

No es gratuito ya que se debe hacer un previo pago para poder adquirir el producto.

Visual Paradigm for UML: es una de las herramientas *UML CASE* del mercado, considerada como muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del *software* que lo automatiza y acelera, permitiendo la captura de requisitos,

análisis, diseño e implementación. Visual *Paradigm for UML* también proporciona características tales como generación de código, ingeniería inversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos, crear la definición de base de datos a partir del esquema de clases. Permite invertir el código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Está diseñada para usuarios interesados en sistemas de *software* de gran escala en el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. (44)

Luego de estudiar las tendencias, tecnologías y herramientas actuales se puede concluir que para el desarrollo del sistema se empleará XP como metodología, por las siguientes características:

El proyecto es pequeño así como el grupo de desarrollo. En el sistema trabajarán 2 desarrolladores. XP está concebida para la realización de proyectos pequeños con un grupo de desarrollo de menos de 20 personas.

Proyecto con alto grado de riesgo. El tiempo de entrega del sistema planteado es corto además los requisitos son muy cambiantes en este tipo de proyectos. XP está diseñado para atenuar los riesgos en estos casos.

Interacción con el usuario final. Esta interacción disminuye el tiempo de comunicación y la cantidad de documentación, junto con los altos costes de su creación y mantenimiento, además se puede lograr un producto que satisfaga las necesidades de los clientes.

Planificación incremental. Como en XP los requisitos varían según las necesidades del negocio, la planificación nunca será lo más acertada posible. El objetivo que se plantea, así como lo es el de la metodología XP es generar pequeñas versiones funcionales del sistema que proporcionen un valor adicional desde el punto de vista del negocio.

Integración continua. En muchos casos la integración de código produce efectos imprevistos, y en ocasiones la integración puede fallar cuando dejan de funcionar cosas por motivos desconocidos. XP plantea que la integración debe ser permanente con el objetivo de que los problemas sean resueltos, evitando así que los defectos se acumulen además conseguir que sean resueltos por los mismos programadores que los introdujeron.

También se empleará como lenguaje de modelado *UML* para modelar el diseño, con una arquitectura MVC debido a que esta separa la interfaz, la lógica de negocio y la presentación. Facilita la realización de pruebas unitarias de los componentes, así como la aplicación del desarrollo guiado por pruebas. Facilita el mantenimiento en caso de errores y ofrece maneras más sencillas para probar el correcto funcionamiento del sistema. Ver figura 1.



Figura 1 Modelo Vista Controlador.

Como lenguaje de programación del lado del cliente HTML y PHP del lado del servidor este último generalmente es utilizado como módulo de *Apache*, lo que lo hace extremadamente veloz. PHP es código abierto, lo cual significa que el usuario no depende de una compañía

específica para arreglar cosas que no funcionan, además no está forzado a pagar actualizaciones anuales para tener una versión que funcione e interactúe con *PostgreSQL*.

Teniendo en cuenta también la posibilidad de emigración de nuestro país al *software* libre se empleará como herramienta Visual Paradigm para el modelado. Se implementará en *NetBeans PHP* como herramienta de desarrollo, porque permite crear aplicaciones *web* con PHP 5 y viene con soporte para Symfony, un gran *framework* MVC escrito en PHP, y que además ha sido escogido como *framework* para el desarrollo de este sistema por su gran documentación, aparte de permitir controlar hasta el último acceso a la información e incluye por defecto protección contra ataques *XSS* (*Cross Site Scripting*) en español (Encriptación de Sitios Cruzada) y *CSRF* (*Cross Site Request Forgery*) en español (Falsificación de Petición en Sitios Cruzados). Su código fuente incluye más de 9000 pruebas unitarias y funcionales. También se empleará *PostgreSQL* como gestor de bases de datos con código fuente disponible libremente. Es uno de los sistemas de gestión de bases de datos de código abierto más potentes y robustos del mercado. Funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Por último *Apache* como servidor *web*, igual por ser de código abierto para plataformas *Unix* y *Windows*, estas herramientas son las seleccionadas porque comparten algo en común que son *software* libre y responden al proceso de emigración de nuestro país.

Conclusiones

En este capítulo se realiza un estudio acerca del estado del arte donde se llega a la conclusión de desarrollar un sistema para la gestión de RRHH de la UERMP. Se tratan una serie de conceptos importantes relacionados con el objetivo del trabajo y se estudian las distintas metodologías y tecnologías existentes en el mundo actual, a la hora de realizar este tipo de aplicaciones.

Capítulo 2: Características del sistema. Fase de exploración y planificación

Introducción

En este capítulo se realiza la valoración de las principales características del sistema a desarrollar, apoyándose en la situación problemática que dio origen al mismo. Se detallan las necesidades de los usuarios, describiéndose las funcionalidades que son objeto de automatización. Se presenta una propuesta del *software* a implementar, especificando detalladamente los requisitos funcionales y no funcionales del sistema.

2.1 Descripción de los procesos vinculados al campo de acción

La solicitud de fuerza laboral se hace por los directores funcionales a través del Modelo de Solicitud de Fuerza de Trabajo. Las solicitudes presentadas serán analizadas y revisadas por el Director de RRHH. Luego de revisada es presentada al Director General quien a su vez se apoya en el Comité de Ingreso, para emitir su decisión final en un término de 5 días, la aprobación o no de la solicitud.

La dirección de RRHH una vez aprobada la solicitud, procede a efectuar la convocatoria de la plaza solicitando en la misma la información preliminar a presentar por los aspirantes para ser analizada por el comité de ingreso acorde a los requisitos del puesto de trabajo. Este valora las características y conocimientos de los aspirantes valiéndose de las técnicas establecidas, a fin de determinar el más idóneo para ocupar la plaza.

Para la entrada del personal a la entidad, se mantendrá un orden de prioridad teniendo en cuenta los siguientes casos:

- ✓ Personal declarado disponible dentro de la organización, que reúna los requisitos establecidos.
- ✓ De no existir personal disponible o aún teniéndolo y no cumplan con los requisitos de la plaza, la Dirección de RRHH procede a efectuar la convocatoria interna, la cual debe permanecer 10 días hábiles, para que accedan a ellas los aspirantes internos.

Capítulo 2: Características del sistema. Fase de exploración y planificación

- ✓ De no existir los internos o los presentados no reúnan los requisitos, se utiliza el servicio de empleo brindado por las Direcciones de Trabajo Municipales, mediante el modelo FT-25 de Solicitud de Fuerza de Trabajo.
- ✓ Como última instancia se realiza la convocatoria externa utilizando los medios de comunicación: la prensa, radio, entre otras.

Una vez realizada la convocatoria, se redacta una carta informando al Sindicato de la entidad sobre su existencia.

La Dirección de RRHH teniendo en cuenta la Resolución 8 del 2005 del Ministerio de Trabajo y Seguridad Social (MTSS), realiza los siguientes pasos:

- ✓ Análisis del puesto de trabajo, definiendo de antemano las competencias laborales de los puestos de trabajo.
- ✓ Informa al Comité de ingreso, de las solicitudes que poseen y las exigencias de cada cargo, así como el resto de la documentación necesaria para que dicho Comité emita su recomendación.
- ✓ Oído el parecer o recomendación del Comité de ingreso, el Director de RRHH, procede a la selección final del candidato.
- ✓ Recibe de los investigadores la información de la verificación de cada uno de los candidatos.

Los aspirantes a la plaza puesta en convocatoria deben presentar los siguientes documentos:

- ✓ Modelo 7 (en caso de ser enviado por la Dirección Municipal de Trabajo).
- ✓ Hoja de Resumen.
- ✓ Carnet de Salud.
- ✓ Actualización del Registro Militar.
- ✓ Carnet de Identidad.

Capítulo 2: Características del sistema. Fase de exploración y planificación

- ✓ Certificación de últimos estudios terminados.
- ✓ Dos fotos en caso de iniciar su vida laboral.
- ✓ Boleta pre empleo.
- ✓ Boleta de asignación del graduado en caso de los egresados de la enseñanza media superior (Técnicos) o superior.

El candidato seleccionado, es acogido desarrollando los siguientes pasos:

- ✓ Presentación al colectivo de trabajadores.
- ✓ Recorrido por las diferentes áreas.
- ✓ Se informa al trabajador sobre la organización y estructura de la empresa, así como su estilo y costumbres.
- ✓ Se le da a conocer el Reglamento Disciplinario Interno de la empresa.
- ✓ Se le brinda información sobre el puesto que va a ocupar.
- ✓ Se le da a conocer sobre los sistemas de estimulación y otras informaciones de interés para el trabajador.

2.2 Objeto de automatización

Durante el ciclo de desarrollo, existen dos procesos que deben ser automatizados puesto que su ejecución de forma manual resulta tediosa y sensible a pérdidas. Estos son de vital importancia para lograr captar entre los candidatos al más apto para ocupar una plaza. Se automatizan los procesos de reclutamiento y selección que realiza la entidad mediante una aplicación *web* que gestione las solicitudes de fuerza laboral realizadas, las convocatorias lanzadas, los candidatos y los trabajadores.

2.3 Propuesta del sistema

El presente trabajo propone la implementación de un sistema *web* que permita establecer el flujo de los procesos de reclutamiento y selección. Entre ellos, gestionar los contenidos de las

Capítulo 2: Características del sistema. Fase de exploración y planificación

solicitudes, las convocatorias, los candidatos a las plazas y los trabajadores. Además de la aprobación de las solicitudes y la selección de los candidatos más idóneos para las plazas a ocupar. También permite realizar una búsqueda filtrada a partir de los parámetros de entrada establecidos.

El sistema es una aplicación *web* y cada una de las funcionalidades se implementa como propósito específico para posterior integración de lo que se denomina “Sistema de Gestión de RRHH en la UERMP”.

2.4 Personas relacionadas con el sistema

Personas relacionadas con el sistema	Justificación
Director de RRHH	Persona que en el sistema gestiona las solicitudes de fuerza laboral, las convocatorias, los candidatos y los trabajadores.
Director funcional	Persona que en el sistema consulta las solicitudes para conocer el estado en que se encuentran.
Director general	Persona que en el sistema aprueba las solicitudes y los candidatos.
Comité de ingreso	Persona que en el sistema consulta los datos de los candidatos aspirantes a la plaza en convocatoria.
Especialista en RRHH	Persona que en el sistema gestiona los candidatos aspirantes a la plaza en convocatoria.
Administrador del sistema	Persona que en el sistema, se encarga de la gestión de los usuarios y los nomencladores.

Tabla 2. 1 Personas relacionadas con el sistema

2.5 Requisitos Funcionales (RF)

RF 1. Autenticar Usuario.

RF 1.1 Iniciar Sesión.

RF 1.2 Cerrar Sesión.

RF 2. Gestionar usuario.

RF 2.1 Insertar usuario.

RF 2.2 Modificar usuario.

RF 2.3 Eliminar usuario.

RF 3. Gestionar solicitud de fuerza laboral.

RF 3.1. Insertar solicitud de fuerza laboral.

RF 3.2. Modificar solicitud de fuerza laboral.

RF 3.3. Consultar solicitud de fuerza laboral.

RF 3.4. Eliminar solicitud de fuerza laboral.

RF 3.5. Aprobar solicitud de fuerza laboral.

RF 4. Gestionar convocatoria.

RF 4.1. Insertar convocatoria.

RF 4.2. Modificar convocatoria.

RF 4.3. Consultar convocatoria.

RF 4.4. Eliminar convocatoria.

RF 5. Gestionar candidato.

RF 5.1. Insertar candidato.

Capítulo 2: Características del sistema. Fase de exploración y planificación

RF 5.2. Modificar candidato.

RF 5.3. Consultar candidato.

RF 5.4. Eliminar candidato.

RF 5.5. Aprobar candidato.

RF 6. Gestionar trabajador.

RF 6.1. Insertar trabajador.

RF 6.2. Modificar trabajador.

RF 6.3. Consultar trabajador.

RF 6.4. Eliminar trabajador.

RF 7. Buscar.

RF 7.1. Buscar solicitud.

RF 7.2. Buscar convocatoria.

RF 7.3. Buscar candidato.

RF 7.4. Buscar trabajador.

RF 7.5. Buscar usuario.

RF 7.6. Buscar rol.

RF 7.8. Buscar permiso.

RF 8. Gestionar rol.

RF 8.1. Insertar rol.

RF 8.2. Modificar rol.

Capítulo 2: Características del sistema. Fase de exploración y planificación

RF 8.3. Eliminar rol.

RF 9. Gestionar área.

RF 9.1. Insertar área.

RF 9.2. Modificar área.

RF 9.3. Eliminar área.

RF 10 Gestionar cargo.

RF 10.1. Insertar cargo.

RF 10.2. Modificar cargo.

RF 10.3. Eliminar cargo.

RF 11 Gestionar grupo escala.

RF 11.1. Insertar grupo escala.

RF 11.2. Modificar grupo escala.

RF 11.3. Eliminar grupo escala.

RF 12 Gestionar categoría ocupacional.

RF 12.1. Insertar categoría ocupacional.

RF 12.2. Modificar categoría ocupacional.

RF 12.3. Eliminar categoría ocupacional.

RF 13 Gestionar nivel escolar.

RF 13.1. Insertar nivel escolar.

RF 13.2. Modificar nivel escolar.

Capítulo 2: Características del sistema. Fase de exploración y planificación

RF 13.3. Eliminar nivel escolar.

RF 14 Mostrar permiso.

2.6 Requisitos No Funcionales (RNF)

Requerimientos de apariencia o interfaz externa

En la aplicación predominará el azul como color que identifica a la entidad. El tamaño de la letra generalmente será de 11 píxeles. Contará con encabezados que indiquen dónde se encuentra el usuario en el sistema. Los nombres de los botones y vínculos deben ser sugerentes para el entendimiento del usuario.

Requerimientos de usabilidad

La aplicación debe ser fácil de navegar sin que se requiera de muchos conocimientos. Debe ser escalable para poder agregar nuevas funcionalidades sin tener que afectar las que ya están funcionando.

Requerimientos de rendimiento

El sistema debe soportar un tiempo de respuesta menor o igual a 5 segundos y soportar una conexión simultánea de más de 100 usuarios.

Requerimiento de portabilidad

Las herramientas utilizadas podrán ser usadas bajo cualquier sistema operativo *Windows NT* en adelante o cualquier distribución de Linux. El servidor *web* y el servidor de base de datos pueden estar en la misma computadora sin ocasionar problema alguno.

Requerimientos de seguridad

Confiabilidad: La información manejada por el sistema debe estar protegida de acceso no autorizado. Identificar al usuario antes de realizar cualquier acción sobre el sistema. Todas las funcionalidades del sistema requerirán que el usuario ya se haya autenticado, en caso contrario el sistema forzará al usuario que lo haga.

Capítulo 2: Características del sistema. Fase de exploración y planificación

Integridad: La información manejada por el sistema debe ser objeto de cuidadosa protección contra la corrupción y estados de inconsistencia. Proteger la información manejada en el sistema de accesos no autorizados. El sistema permitirá que el usuario no tenga contacto con información que no sea manejada por él. La base de datos posee sus propias tablas, por lo que no se corre el riesgo de que se pierda la información.

Disponibilidad: La aplicación deberá estar disponible en todo momento para aquellas personas con acceso a la información, y los mecanismos utilizados para lograr la seguridad, no deben ser un obstáculo.

Requerimientos de software

En el servidor de aplicaciones estará instalado *Apache 2.0* y en el servidor de base de datos, *PostgreSQL 8.4* o superior. Para que funcione la aplicación, se utilizará los navegadores *Internet Explorer 7.0* o superior, o *Mozilla Firefox*.

Requerimientos de hardware

En el cliente se requiere una máquina con 256 MB de RAM como mínimo, un microprocesador Pentium 4 a más de 1.6 GHz y un disco duro de más de 10 GB, el servidor *web* junto con el servidor de base de datos debe tener al menos 512 MB de RAM y 20 GB de disco duro mínimo, todas las máquinas implicadas en la funcionalidad de la aplicación deben estar conectadas a la red de al menos 100 Mbps de velocidad.

2.7 Fase de exploración

2.7.1 Historias de usuario (HU)

Historia de Usuario	
Número: 1	Usuario: Director de RRHH, director funcional, director general, especialista en RRHH, comité de ingreso y administrador del sistema.
Nombre historia: Autenticar usuario.	

Capítulo 2: Características del sistema. Fase de exploración y planificación

Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Diego Quesada y Lilianne Reyes	
Descripción: Permite la autenticación de los usuarios en el sistema.	
Observaciones:	

Tabla 2. 2 HU 1 Autenticar usuario

Historia de Usuario	
Número: 2	Usuario: Administrador del sistema.
Nombre historia: Gestionar usuario.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Diego Quesada y Lilianne Reyes	
Descripción: Permite insertar, modificar y eliminar los usuarios que interactúan con el sistema.	
Observaciones:	

Tabla 2. 3 HU 2 Gestionar usuario

Historia de Usuario	
Número: 3	Usuario: Director de RRHH, director general y director funcional.
Nombre historia: Gestionar solicitud de fuerza laboral.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1.5	Iteración asignada: 2
Programador responsable: Diego Quesada y Lilianne Reyes	

Capítulo 2: Características del sistema. Fase de exploración y planificación

Descripción: Permite insertar, modificar, consultar, aprobar y eliminar una solicitud de fuerza laboral.
Observaciones:

Tabla 2. 4 HU 3 Gestionar solicitud de fuerza laboral

Historia de Usuario	
Número: 4	Usuario: Director de RRHH.
Nombre historia: Gestionar convocatoria.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1.5	Iteración asignada: 2
Programador responsable: Diego Quesada y Lilianne Reyes	
Descripción: Permite insertar, modificar, consultar y eliminar una convocatoria de una plaza.	
Observaciones:	

Tabla 2. 5 HU 4 Gestionar convocatoria

Historia de Usuario	
Número: 5	Usuario: Especialista en RRHH y comité de ingreso
Nombre historia: Gestionar candidato.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1.5	Iteración asignada: 3
Programador responsable: Diego Quesada y Lilianne Reyes	
Descripción: Permite insertar, modificar, consultar, aprobar y eliminar un candidato aspirante a una plaza en convocatoria.	
Observaciones:	

Tabla 2. 6 HU 5 Gestionar candidato

Capítulo 2: Características del sistema. Fase de exploración y planificación

Historia de Usuario	
Número: 6	Usuario: Director de RRHH
Nombre historia: Gestionar trabajador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1.5	Iteración asignada: 3
Programador responsable: Diego Quesada y Lilianne Reyes	
Descripción: Permite insertar, modificar, consultar y eliminar un trabajador.	
Observaciones:	

Tabla 2. 7 HU 6 Gestionar trabajador

Historia de Usuario	
Número: 7	Usuario: Director de RRHH, director general, director funcional, especialista de RRHH, comité de ingreso y administrador del sistema
Nombre historia: Buscar.	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Diego Quesada y Lilianne Reyes	
Descripción: Permite buscar solicitudes, convocatorias, candidatos, trabajadores, usuarios, roles y permisos según parámetros de búsquedas.	
Observaciones:	

Tabla 2. 8 HU 7 Buscar

Historia de Usuario	
Número: 8	Usuario: Administrador del sistema.

Capítulo 2: Características del sistema. Fase de exploración y planificación

Nombre historia: Gestionar rol	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Diego Quesada y Lilianne Reyes	
Descripción: Permite insertar, modificar y eliminar un rol del sistema.	
Observaciones:	

Tabla 2. 9 HU 8 Gestionar rol

Historia de Usuario	
Número: 9	Usuario: Administrador del sistema.
Nombre historia: Gestionar área	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Diego Quesada y Lilianne Reyes	
Descripción: Permite insertar, modificar y eliminar un área.	
Observaciones:	

Tabla 2. 10 HU 9 Gestionar área

Historia de Usuario	
Número: 10	Usuario: Administrador del sistema
Nombre historia: Gestionar cargo	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1

Capítulo 2: Características del sistema. Fase de exploración y planificación

Programador responsable: Diego Quesada y Lilianne Reyes
Descripción: Permite insertar, modificar y eliminar un cargo.
Observaciones:

Tabla 2. 11 HU 10 Gestionar cargo

Historia de Usuario	
Número: 11	Usuario: Administrador del sistema.
Nombre historia: Gestionar grupo escala	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Diego Quesada y Lilianne Reyes	
Descripción: Permite insertar, modificar y eliminar un grupo escala.	
Observaciones:	

Tabla 2. 12 HU 11 Gestionar grupo escala

Historia de Usuario	
Número: 12	Usuario: Administrador del sistema
Nombre historia: Gestionar categoría ocupacional	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Diego Quesada y Lilianne Reyes	
Descripción: Permite insertar, modificar y eliminar una categoría ocupacional.	
Observaciones:	

Tabla 2. 13 HU 12 Gestionar categoría ocupacional

Capítulo 2: Características del sistema. Fase de exploración y planificación

Historia de Usuario	
Número: 13	Usuario: Administrador del sistema.
Nombre historia: Gestionar nivel escolar	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Diego Quesada y Lilianne Reyes	
Descripción: Permite insertar, modificar y eliminar un nivel escolar.	
Observaciones:	

Tabla 2. 14 HU 13 Gestionar nivel escolar

Historia de Usuario	
Número: 14	Usuario: Administrador del sistema.
Nombre historia: Mostrar permiso	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Diego Quesada y Lilianne Reyes	
Descripción: Permite mostrar los permiso existentes en el sistema.	
Observaciones:	

Tabla 2. 15 HU 14 Mostrar permiso

2.8 Fase de Planificación

2.8.1 Planificación de las Historias de Usuarios

Para el buen desarrollo del sistema propuesto, se realizó una estimación de cada una de las historias de usuario identificadas, la cual arrojó los resultados que se muestran a continuación:

Capítulo 2: Características del sistema. Fase de exploración y planificación

No	Nombre de la HU	Prioridad	Riesgo	Esfuerzo (días)	Iteración
1	Autenticar usuario	Media	Bajo	1	1
2	Gestionar usuario	Alta	Bajo	4	1
3	Gestionar solicitud de fuerza laboral	Alta	Medio	9	2
4	Gestionar convocatoria	Alta	Medio	9	2
5	Gestionar candidato	Alta	Medio	9	3
6	Gestionar trabajador	Alta	Medio	9	3
7	Buscar	Media	Bajo	2	3
8	Gestionar rol	Media	Bajo	1	1
9	Gestionar área	Media	Bajo	1	1
10	Gestionar cargo	Media	Bajo	1	1
11	Gestionar grupo escala	Media	Bajo	1	1
12	Gestionar categoría ocupacional	Media	Bajo	1	1
13	Gestionar nivel escolar	Media	Bajo	1	1
14	Mostrar permiso	Baja	Bajo	1	1

Tabla 2. 16 Estimación de esfuerzos por historias de usuarios

2.8.2 Plan de Iteraciones

Después de haber sido identificadas y definidas las historias de usuario y haber hecho un estudio de la estimación propuesta para la construcción de las mismas, se procede a la planificación de la etapa de implementación, en la cual se van a definir cuales historias de usuario se implementarán por cada iteración definida, así como las posibles fechas de entrega de cada una de ellas.

Capítulo 2: Características del sistema. Fase de exploración y planificación

Este paso es de suma importancia en el ciclo de desarrollo de XP. Las funcionalidades son planificadas en esta fase, generando al final de cada una, un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.

En el caso del sistema actual se definieron cuatro iteraciones para proceder con las historias de usuarios precisadas. A continuación se detallan como estará estructurada cada una de estas.

Iteración 1: Se propone codificar las historias de usuario, que proveen las funcionalidades del sistema:

- ✓ HU 1 Autenticar usuario.
- ✓ HU 2 Gestionar usuario.
- ✓ HU 8 Gestionar rol.
- ✓ HU 9 Gestionar área.
- ✓ HU 10 Gestionar cargo.
- ✓ HU 11 Gestionar grupo escala.
- ✓ HU 12 Gestionar categoría ocupacional.
- ✓ HU 13 Gestionar nivel escolar.
- ✓ HU 14 Mostrar permiso.

Iteración 2: Se codifican las historias de prioridad alta y riesgo medio que requieren de mucha interacción con el usuario:

- ✓ HU 3 Gestionar solicitud de fuerza laboral.
- ✓ HU 4 Gestionar convocatoria”.

Capítulo 2: Características del sistema. Fase de exploración y planificación

Iteración 3: Se continúa la codificación de las historias que requieren de mucha interacción con el usuario:

- ✓ HU 5 Gestionar candidato.
- ✓ HU 6 Gestionar trabajador.
- ✓ HU 7 Buscar.

2.8.3 Plan de duración de las iteraciones

Siguiendo el desarrollo de la metodología XP se crea el plan de duración de las iteraciones. En este plan se especifica más detalladamente el orden de desarrollo de las historias de usuarios dentro de cada iteración así como la estimación completa de dicha iteración.

Iteraciones	Orden de las HU de usuario a implementar	Duración total de las iteraciones
Iteración 1	Autenticar usuario. Gestionar usuario. Gestionar rol. Gestionar área. Gestionar cargo. Gestionar grupo escala. Gestionar categoría ocupacional. Gestionar nivel escolar. Mostrar permiso.	2 Semanas.
Iteración 2	Gestionar solicitud de fuerza laboral. Gestionar convocatoria.	3 Semanas.
Iteración 3	Gestionar candidato.	

Capítulo 2: Características del sistema. Fase de exploración y planificación

	Gestionar trabajador Buscar	3 Semanas
--	--------------------------------	-----------

Tabla 2. 17 Plan de duración de las iteraciones

2.8.4 Plan de entregas

Seguidamente se presenta el plan de entregas elaborado para la fase de implementación. El mismo se agrupa por *releases* (liberación) del sistema al finalizar cada una de las iteraciones, con la fecha de estimación de terminación de cada una.

Módulos	Final de la 1ra. Iteración 3ra. semana de abril	Final de la 2da. Iteración 2da. semana de mayo	Final de la 3ra. Iteración 4ta. semana de mayo
Reclutamiento y Selección	0.1	0.2	1.0

Tabla 2. 18 Plan de duración de entregas

Luego del análisis de los flujos de trabajo actuales se obtiene un listado de las funcionalidades que debe tener el sistema. Se definen y redactan las historias de usuario así como la planificación de las mismas, se estima un tiempo ideal de 50 días necesario para implementar la herramienta. Partiendo de estos puntos, se puede comenzar con la construcción de la propuesta, siempre velando que cumpla con las características antes precisadas.

Conclusiones

Durante el desarrollo de este capítulo además de señalar las funcionalidades que tendrá la aplicación propuesta, se hace referencia a todo lo relacionado a las fases de exploración y planificación del sistema, realizando una descripción de cada uno de los artefactos generados durante las mismas.

Capítulo 3: Implementación y Pruebas

Introducción

La implementación de *software* en XP es un proceso que se realiza de forma iterativa, obteniendo como resultado de cada una de estas, un producto funcional que debe ser sometido a pruebas y mostrado al cliente para permitir una retroalimentación por parte de los desarrolladores. El siguiente capítulo está dedicado a detallar las 4 iteraciones llevadas a cabo durante la etapa de construcción del sistema, exponiéndose cada una de las tareas designadas por historia de usuarios. De igual modo se exponen las pruebas de aceptación efectuadas sobre el proyecto.

3.1 Diseño del sistema

3.1.1 Tarjetas Clase, Responsabilidad y Colaboración (CRC)

Para el diseño de las aplicaciones, la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC. No obstante, el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante.

<i>sfGuardAuth</i>	
Responsabilidad	Colaboraciones
Iniciar sesión Cerrar sesión	<i>sfGuardUser</i> <i>sfGuardGroup</i> <i>sfGuardPermission</i>

Tabla 3. 1 Tarjeta CRC clase *sfGuardAuth*

<i>sfGuardUser</i>	
Responsabilidad	Colaboraciones

Insertar usuario.	<i>sfGuardAuth</i>
Modificar usuario.	<i>sfGuardGroup</i>
Eliminar usuario.	<i>sfGuardPermission</i>

Tabla 3. 2 Tarjeta CRC clase *sfGuardUser*

Solicitud	
Responsabilidad	Colaboraciones
Insertar solicitud	Área
Modificar solicitud	Cargo
Consultar solicitud	Categoría_ocupacional
Eliminar solicitud	Grupo_escala
Aprobar solicitud	<i>sfGuardAuth</i>
	<i>sfGuardGroup</i>
	<i>sfGuardPermission</i>

Tabla 3. 3 Tarjeta CRC clase Solicitud

Convocatoria	
Responsabilidad	Colaboraciones
Insertar convocatoria	Área.
Modificar convocatoria	Cargo.
Consultar convocatoria	<i>sfGuardAuth</i>
Eliminar convocatoria	<i>sfGuardGroup</i>
	<i>sfGuardPermission</i>

Tabla 3. 4 Tarjeta CRC clase Convocatoria

Candidato	
Responsabilidad	Colaboraciones
Insertar candidato.	Cargo.
Modificar candidato	Nivel_escolar.
Consultar candidato	<i>sfGuardAuth</i>
Eliminar candidato.	<i>sfGuardGroup</i>
Aprobar candidato.	<i>sfGuardPermission</i>

Tabla 3. 5 Tarjeta CRC clase Candidato

Trabajador	
Responsabilidad	Colaboraciones
Insertar datos trabajador	Área.
Modificar datos trabajador	Cargo.
Consultar datos trabajador	Categoría_ocupacional.
Eliminar trabajador	Nivel_escolar
	<i>sfGuardAuth</i>
	<i>sfGuardGroup</i>
	<i>sfGuardPermission</i>

Tabla 3. 6 Tarjeta CRC clase Trabajador

Buscar	
Responsabilidad	Colaboraciones
Buscar solicitud	Área

Buscar convocatoria	Cargo
Buscar candidato	<i>sfGuardAuth</i>
Buscar trabajador	<i>sfGuardGroup</i>
Buscar usuario	<i>sfGuardPermission</i>
Buscar rol	
Buscar permiso	

Tabla 3. 7 Tarjeta CRC clase Buscar

Área	
Responsabilidad	Colaboraciones
Insertar área	<i>sfGuardAuth</i>
Modificar área	<i>sfGuardGroup</i>
Eliminar área	<i>sfGuardPermission</i>

Tabla 3. 8 Tarjeta CRC clase Área

Cargo	
Responsabilidad	Colaboraciones
Insertar cargo	<i>sfGuardAuth</i>
Modificar cargo	<i>sfGuardGroup</i>
Eliminar cargo	<i>sfGuardPermission</i>

Tabla 3. 9 Tarjeta CRC clase Cargo

Rol	
Responsabilidad	Colaboraciones

Insertar rol	<i>sfGuardAuth</i>
Modificar rol	<i>sfGuardGroup</i>
Eliminar rol	<i>sfGuardPermission</i>

Tabla 3. 10 Tarjeta CRC clase Rol

Grupo_escala	
Responsabilidad	Colaboraciones
Insertar grupo escala	<i>sfGuardAuth</i>
Modificar grupo escala	<i>sfGuardGroup</i>
Eliminar grupo escala	<i>sfGuardPermission</i>

Tabla 3. 11 Tarjeta CRC clase Grupo_escala

Categoría_ocupacional	
Responsabilidad	Colaboraciones
Insertar categoría ocupacional	<i>sfGuardAuth</i>
Modificar categoría ocupacional	<i>sfGuardGroup</i>
Eliminar categoría ocupacional	<i>sfGuardPermission</i>

Tabla 3. 12 Tarjeta CRC clase Categoría_ocupacional

Nivel_escolar	
Responsabilidad	Colaboraciones
Insertar nivel escolar	<i>sfGuardAuth</i>
Modificar nivel escolar	<i>sfGuardGroup</i>
Eliminar nivel escolar	<i>sfGuardPermission</i>

Tabla 3. 13 Tarjeta CRC clase Nivel_escolar

Permiso	
Responsabilidad	Colaboraciones
Mostrar permiso	<i>sfGuardAuth</i> <i>sfGuardGroup</i>

Tabla 3. 14 Tarjeta CRC clase Permiso

3.1.2 Modelo de datos

El diseño de la base de datos (BD) es uno de los pasos más importantes durante el ciclo de desarrollo del sistema, debido a que uno de sus objetivos fundamentales es brindar persistencia al modelo de diseño elaborado.

La base de datos diseñada cuenta con un total de 15 tablas, de ellas 9 son básicas en el negocio y 6 para la gestión de usuarios. Estas últimas fueron creadas con la instalación del *plugin* (componente) de Symfony *sfGuardPlugin* para la gestión de usuarios. Ver Anexo 1.

3.2 Fase de Implementación

3.2.1 Tareas de Ingeniería

Las tareas de ingeniería no son más que un conjunto de acciones a desarrollar para resolver las historias de usuario. Permiten organizar el proceso de implementación además de posibilitar que sea conocido el grado de complejidad de cada historia de usuario teniendo en cuenta la cantidad de tareas asociadas a ella.

Iteración 1

En la primera iteración se debe implementar las características principales del sistema, para poder liberar una primera versión o prototipo funcional al cliente. Las HU 1, 2, 8, 9, 10, 11, 12, 13 y 14 fueron las seleccionadas para realizar en la primera iteración.

Para ver las tareas correspondientes a la iteración 1. Ver Anexo 2.

Iteración 2

En la segunda iteración se implementan dos HU. Estas se dividieron en cuatro tareas de la ingeniería fundamentales, que proveen las funcionalidades básicas de insertar, modificar, consultar y eliminar, además de aprobar solicitud.

Para ver las tareas correspondientes a la iteración 2. Ver Anexo 3.

Iteración 3

En la tercera iteración igual que en la iteración anterior se implementarán tres historias de usuario que interactúan directamente con el mismo. Las HU 5, 6 y 7 fueron las seleccionadas para realizar en esta iteración.

Para ver las tareas correspondientes a la iteración 3. Ver Anexo 4.

3.3 Diagrama de Despliegue

El Diagrama de Despliegue se utiliza para modelar el *hardware* utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Esto muestra la configuración de los elementos de *hardware* (nodos) y muestra cómo los elementos y artefactos del *software* se trazan en esos nodos. El sistema está distribuido por una estación de trabajo cliente que se comunica mediante HTTP a un servidor *web* que tiene instalado *Apache*. Este a su vez se comunica por TCP/IP con un servidor de base de datos que tiene instalado *PostgreSQL*.

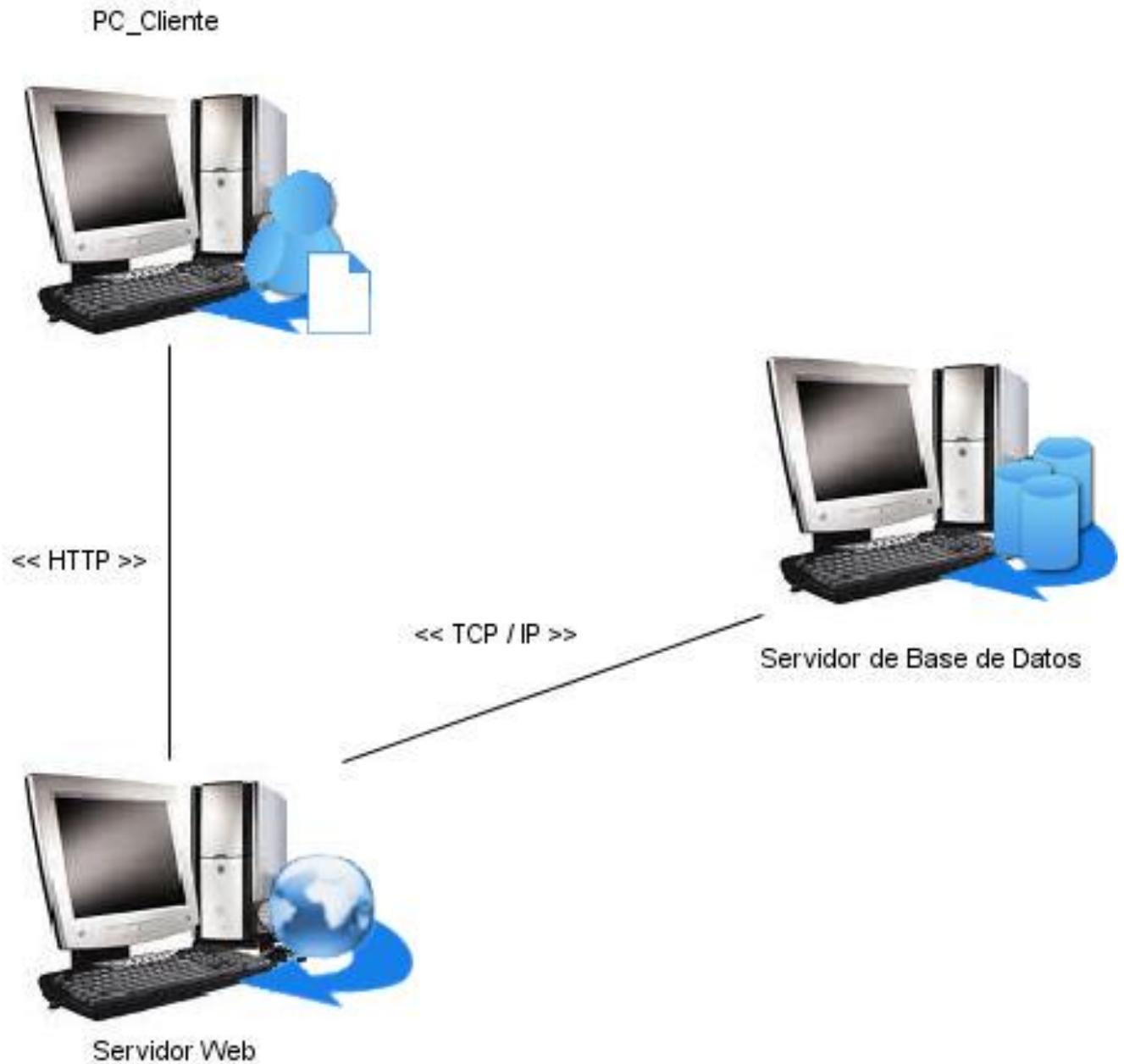


Figura 2 Diagrama de despliegue.

3.4 Fase de Pruebas

El proceso de pruebas es uno de los pilares de la Programación Extrema. XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos

colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

A tono con la metodología XP, Symfony propone la realización de pruebas con el fin de asegurar la calidad de la futura aplicación. No obstante, y aún cuando la esencia continúa siendo la misma, las pruebas recomendadas por Symfony son pruebas automatizadas. Este tipo de pruebas, constituyen uno de los mayores avances en la programación desde la aparición de la orientación a objetos y aunque son recomendadas, su uso o no, queda a merced de los desarrolladores. Para el desarrollo de las pruebas del presente trabajo se realiza una combinación de pruebas manuales con pruebas automatizadas, considerando que ambos tipos se complementan de manera eficiente logrando así una mayor calidad de la aplicación.

3.4.1 Pruebas de aceptación

Las pruebas funcionales son la mejor forma de probar la aplicación de extremo a extremo: desde la petición realizada por un navegador hasta la respuesta enviada por el servidor. Las pruebas funcionales prueban todas las capas de la aplicación: el sistema de enrutamiento, el modelo, las acciones y las plantillas. En realidad, son muy similares a lo que se hace manualmente cada vez que se añade o modifica una acción y se prueban dichos cambios en el navegador para comprobar que todo funciona bien al pulsar sobre los enlaces y botones y que todos los elementos se muestran correctamente en la página. En otras palabras, lo que se hace es probar un escenario correspondiente a la historia de usuario que se acaba de implementar en la aplicación.

Para ver las pruebas realizadas en la aplicación. Ver Anexo 5.

Por cada iteración realizada se le entrega al cliente una liberación del producto con una cierta cantidad de funcionalidades propuestas, donde este verifica su aceptación o no con las

funcionalidades liberadas en cada etapa. El siguiente gráfico muestra el nivel de aceptación que tuvo el cliente en cada una de las iteraciones.

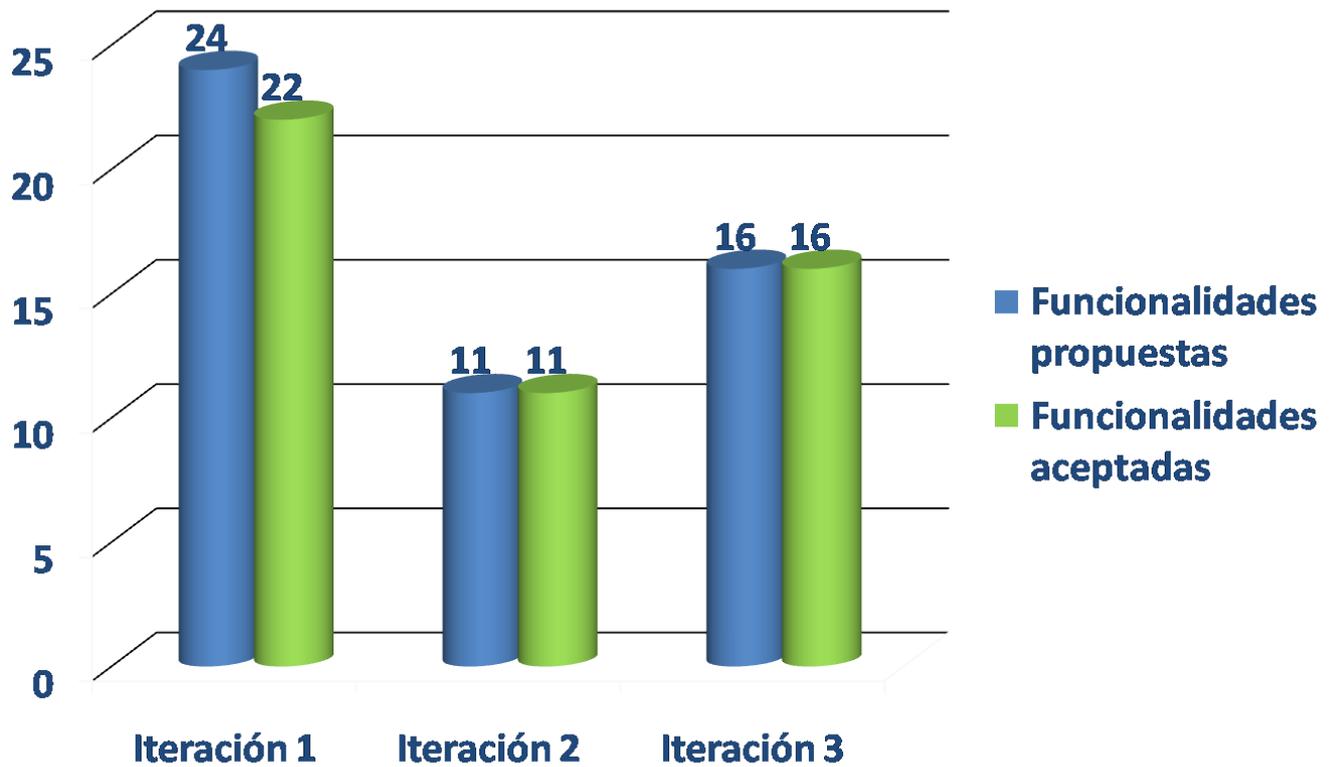


Figura 3 Gráfico de prueba de aceptación.

Conclusiones

En este capítulo se describen las tarjetas CRC, las cuales brindan claridad en aspectos como las principales funcionalidades que presentan las clases así como la relación existente entre ellas. Por otro lado, se diseña el modelo de datos encargado de brindar soporte a la aplicación en cuestión. Otro de los artefactos generados fue las tareas por historia de usuarios, el mismo es de vital importancia por cuanto ayudan a los desarrolladores a profundizar más en los aspectos a tener en cuenta a la hora de implementar las HU detectadas. Además se describe el proceso de prueba, uno de los más importantes para garantizar el éxito de la aplicación propiciando al cliente conformidad y seguridad ante el sistema. Con el fin de este capítulo se da por terminada la propuesta que trae este trabajo.

Capítulo 4: Estudio de Factibilidad

Introducción

Una de las principales tareas dentro de la planificación de proyectos de *software* es la estimación, la cual consiste en determinar, los recursos de *hardware* y *software*, el costo, el tiempo y esfuerzo, necesarios para el desarrollo de los mismos. Este capítulo pretende realizar un estudio de factibilidad correspondiente al sistema propuesto, realizándose una estimación del esfuerzo necesario para llevar a cabo el mismo.

4.1 Características del Proyecto

Nombre de la entrada externa	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Autenticar usuario	3	2	Simple
Gestionar usuario	3	7	Simple
Gestionar solicitud de fuerza laboral	5	8	Simple
Gestionar convocatoria	4	7	Simple
Gestionar candidato	3	9	Simple
Gestionar trabajador	5	11	Simple
Gestionar rol	2	3	Simple
Gestionar área	1	1	Simple
Gestionar cargo	1	1	Simple
Gestionar grupo escala	1	1	Simple
Gestionar categoría ocupacional	1	1	Simple
Gestionar nivel escolar	1	1	Simple

Capítulo 4: Estudio de Factibilidad

Total		52	
--------------	--	-----------	--

Tabla 4. 1 Entradas externas.

Nombre de la salida externa	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Gestionar solicitud de fuerza laboral	5	8	Simple
Gestionar convocatoria	4	7	Simple
Gestionar candidato	3	9	Simple
Gestionar trabajador	5	11	Simple
Buscar	1	2	Simple
Total		37	

Tabla 4. 2 Salidas externas.

Nombre de la consulta externa	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Total		0	

Tabla 4. 3 Consultas externas.

Nombre de los archivos lógicos internos	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Solicitud de fuerza laboral	5	8	Simple
Convocatoria	4	7	Simple
Candidato	3	9	Simple

Trabajador	5	11	Simple
Área	1	1	Simple
Cargo	1	1	Simple
Categoría ocupacional	1	1	Simple
Grupo escala	1	1	Simple
Nivel escolar	1	1	Simple
Usuario	3	7	Simple
Total		47	

Tabla 4. 4 Archivos lógicos internos.

Nombre de los archivos de interfaz externos	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Total		0	

Tabla 4. 5 Archivos de interfaz externos.

Estimación Inicial

Puntos de Función sin ajustar (UFP): se obtiene a través de la suma del aporte de todos los elementos.

Elementos	Baja		Media		Alta		Aportes
	Cant.	Valor	Cant.	Valor	Cant.	Valor	
Entradas externas	12	3	0	4	0	6	36
Salidas externas	5	4	0	5	0	7	20
Consultas externas	0	3	0	4	0	6	0

Archivos lógicos internos	10	7	0	10	0	15	70
Archivos de interfaz externos	0	5	0	7	0	10	0
Total							126

Tabla 4. 6 Puntos de Función desajustados.

UFP= 126

4.2 Cálculo de instrucciones fuertes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo

Dado la obtención de los Puntos de Función sin ajustar del sistema se puede estimar el esfuerzo, para esto se utiliza el método COCOMO II.

El método COCOMO II consiste en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar estimados para un proyecto. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo que influyen en el esfuerzo requerido para el desarrollo del *software*. La meta es obtener un número que caracterice completamente al sistema.

4.2.1 Cálculo del esfuerzo nominal

Para el cálculo del esfuerzo nominal se utiliza la ecuación:

$$PM \text{ nominal} = A * (\text{Size})^E$$

PM nominal: es el esfuerzo nominal requerido en mes-hombre.

A: Es una constante que se utiliza para capturar los efectos multiplicativos en el esfuerzo requerido de acuerdo al crecimiento del tamaño del *software*. El modelo la calibra con un valor de 2.94.

Size: Es el tamaño estimado del *software*, en Puntos de Función sin ajustar (convertibles a KSLOC). Se calcula el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema. Se utiliza PHP (factor de conversión = 53 SLOC/UFP).

$$\text{Size} = 53 * 126 = 6678 \text{ SLOC}$$

$$\text{Size} = 6.678 \text{ KSLOC}$$

E: Es una constante, denominada factor escalar. Se calcula ponderando las variables escalares, mediante la ecuación:

$$E = 0.91 + 0.01 * \sum (W_i)$$

W_i = valor de la variable escalar

Donde las W_i se muestran en la siguiente tabla

Variable	Descripción	Valor
PREC	El sistema es muy familiar	2.25
FLEX	Algo de relajación en cuanto a la flexibilidad del desarrollo	2.50
RESL	La arquitectura es sólida y los riesgos generalmente se mitigan	3.50
TEAM	La interacción del equipo es altamente cooperativa	2.80
PMAT	La madurez del proceso <i>software</i> es baja	1.80
Total		12.85

Tabla 4. 7 Factor Escalar (W_i).

$$E = 0.91 + 0.01 * 12.85 = 1.0385$$

Por tanto:

$$\begin{aligned} PM \text{ nominal} &= A * (\text{Size})^E = 2.94 * (6.678 \text{ KSLOC})^{1.0385} \\ &= 21.122369329770562930127448373347 \approx 21.12 \text{ mes} - \text{hombre} \end{aligned}$$

4.2.2 Cálculo del esfuerzo ajustado

Aplicando la ecuación de esfuerzo ajustado:

Capítulo 4: Estudio de Factibilidad

$$\text{PM ajustado} = \text{PM nominal} * \prod(\text{ME}_i)$$

Donde los Multiplicadores de Esfuerzo (ME) son mostrados en la siguiente tabla:

Multiplicador	Descripción	Valor
PERS	Se tienen analistas y programadores con alta eficiencia y capacidad de trabajo en equipo.	0.90
RCPX	Las exigencias de confiabilidad, documentación y volumen de datos son moderadas, y la complejidad del producto es baja.	1.05
RUSE	No se pretende reutilizar nada	1.07
PDIF	No existen restricciones en cuando al tiempo de CPU (unidad central de procesamiento) o al consumo de memoria, la plataforma es muy estable.	1.00
PREX	Tanto los analistas como los programadores tienen aproximadamente 6 meses de experiencia en la aplicación, la plataforma, el lenguaje y las herramientas utilizadas.	0.95
SCED	Se requiere terminar el proyecto en el tiempo estimado.	1.00
FCIL	Se tienen herramientas CASE simples e infraestructura de comunicaciones básica	0.95
Total		0.96

Tabla 4. 8 Multiplicadores de esfuerzo (ME).

Se arriba al siguiente resultado:

$$\prod(\text{ME}_i) = 0.96$$

$$\text{PM ajustado} = 21.12 * 0.96 = 20.2752 \approx 20.28 \text{ mes} - \text{ hombre}$$

4.2.3 Cálculo del tiempo de desarrollo, cantidad de hombres y costo

Valores calibrados:

$$A = 2.94; B = 0.91; C = 3.67; D = 0.24$$

$$F = D + 0.2 * (E - B) = 0.24 + 0.2 * (1.0385 - 0.91) = 0.2657$$

$$TDEV (\text{Tiempo de desarrollo}) = C * (PM \text{ ajustado})^F = 3.67 * 20.28^{0.2657}$$

$$= 8,1649591965820033671205953349424 \text{ meses} \approx 8.16 \text{ meses}$$

$$CH (\text{Cantidad de Hombres}) = PM_{\text{ajustado}} / TDVE$$

$$= 20.28 / 8.16 = 2.4852941176470588235294117647059 \approx 3 \text{ personas}$$

Como la cantidad real de hombres disponibles para el desarrollo de la aplicación es 2, al reajustar el tiempo de desarrollo según la cantidad de hombres, resulta un tiempo equivalente a 10.14 \approx 10 meses.

Salario promedio: Para determinar el salario promedio se tuvo en cuenta que los desarrolladores son estudiantes de quinto año, por lo que se toma como salario correspondiente \$100.00

$$\text{Costo} = CH * \text{Salario} * PM \text{ ajustado} = 2 * 100 * 20.28 = \$ 4056$$

En la siguiente tabla se presentan los resultados de los cálculos efectuados:

Cálculo de:	Valor
Esfuerzo	20.28 meses-hombre
Tiempo de desarrollo	10 meses
Cantidad de hombres	2 hombres
Salario medio	\$ 100
Costo	\$ 4056

Tabla 4. 9 Resultados.

4.3 Beneficios tangibles e intangibles

Se puede destacar que con el sistema propuesto se logra responder con rapidez a las diversas consultas generando información confiable, precisa y oportuna. Además de lograr una reducción considerable del tiempo al poder manejar los datos de forma automatizada permitiendo insertar o modificar los mismos.

El sistema tiene como beneficios que reduce considerablemente la duplicación de información, la acumulación en papel de la misma, y evita además la pérdida de documentos. Problemas que son muy comunes actualmente dentro de la empresa a la cual se le desarrolla esta aplicación. El *software* que se propone, evita el desgaste físico que provoca generar manualmente la información de este proceso de reclutamiento y selección, así como las dificultades que puede crear no hacerlo de manera correcta, al proveer a los usuarios de un servicio de captura automática que sigue las planificaciones almacenadas en la base de datos.

4.4 Análisis del costo

Asociado al desarrollo de un producto existe siempre un costo de producción, el cual debe ser justificado, fundamentalmente en base a los beneficios que el mismo ha de reportar. El sistema propuesto no constituye un gasto considerable pues todas las herramientas que se han empleado en su desarrollo son libres y de código abierto. El sistema está orientado al trabajador y es de fácil aprendizaje. Además, no son necesarios los gastos por concepto de tecnología pues la institución cuenta con la infraestructura necesaria para la implantación de la aplicación.

Conclusiones

En el presente capítulo se ha realizado un análisis de factibilidad de la solución propuesta, la cual ha determinado la viabilidad de su desarrollo, basada en la comparación entre los costos de su producción y desarrollo, contra los beneficios reportados con su puesta en funcionamiento. En otras palabras el estudio realizado, ha proporcionado valiosos argumentos que permiten llegar a la conclusión de que la solución de *software*, es factible pues reportará importantes beneficios sin incurrir en mayores gastos.

Conclusiones generales

Con el desarrollo del presente trabajo se logra una confiabilidad, integridad y disponibilidad de los datos almacenados, lo cual no podía garantizarse anteriormente.

Se obtienen los resultados esperados en las pruebas unitarias y de aceptación realizadas, logrando que al cliente le resulten satisfactorias las funcionalidades de la aplicación.

Se despliega la aplicación, obteniendo resultados satisfactorios en la explotación del producto.

Con la implementación de la aplicación se da cumplimiento al objetivo propuesto ya que se automatizan los procesos de reclutamiento y selección, con el Sistema de Gestión de RRHH de la UERMP teniendo en cuenta las tareas planteadas.

Se realiza un estudio de factibilidad de la solución en cuestión donde se concluye que la solución propuesta es factible, pues reporta importantes beneficios económicos sin incurrir en mayores gastos.

Recomendaciones

Se recomienda continuar el desarrollo del sistema, en base al incremento de los requisitos del cliente.

Se recomienda trabajar en base a obtener una conexión segura (HTTPS).

Bibliografía Referenciada

1. La *web* de los recursos humanos y el empleo. [En línea] [Citado el: 15 de Octubre de 2009.] <http://www.rrhh-web.com/introduccion.html>.
2. **Ponjuán, Gloria.** Gestión de la información en las organizaciones. *Principios, Conceptos y Aplicaciones. "La Era de la Información"*. 2008.
3. DeGerencia.com. [En línea] [Citado el: 16 de Octubre de 2009.] http://www.degerencia.com/tema/recursos_humanos.
4. **Escat, María.** GestioPolis.com. [En línea] [Citado el: 17 de Octubre de 2009.] <http://www.gestiopolis.com/canales/derrhh/articulos/37/grrhhest.htm>.
5. **Pressman, Roger.** *Ingeniería del Software. Un Enfoque Práctico*. La Habana : Félix Varela, 2007.
6. RRHHWeb. [En línea] [Citado el: 20 de Octubre de 2009.] http://www.rrhhweb.com/includes_contenido/beneficios.html.
7. NexDigital. [En línea] [Citado el: 22 de Octubre de 2009.] http://www.nexo-digital.com/humanex---software-para-la-gestion-de-procesos-de-seleccion-y-ofertas---21-fi.asp?ver_info=1.
8. GrupoCastilla. [En línea] [Citado el: 23 de Octubre de 2009.] <http://www.grupocastilla.es/Noticiasyeventos/Noticias/Noticias2009/ResultadosyEstrategiadeGrupoCastilla/tabid/138/Default.aspx>.
9. **López, Sergio.** ComputerWorld. [En línea] [Citado el: 25 de Octubre de 2009.] <http://www.idg.es/computerworld/articulo.asp?id=185579>.
10. Assets.Sistema de Gestión Integral. [En línea] [Citado el: 26 de Octubre de 2009.] <http://assets.co.cu/humanos.asp>.
11. Assets.Sistema de Gestión Integral. [En línea] [Citado el: 26 de Octubre de 2009.] <http://assets.co.cu/clientes.asp?page=15>.

12. HavaSoft. [En línea] [Citado el: 27 de Octubre de 2009.]
<http://www.havasoft.minaz.cu/node/30>.
13. Rational Unified Process. [En línea] [Citado el: 1 de Noviembre de 2009.]
<http://www.rational.com.ar/herramientas/rup.html>.
14. Tech Reports. [En línea] [Citado el: 2 de Noviembre de 2009.]
<http://neocygnus2.blogspot.com/2004/07/cambio-metodologico-rup-xp-1-parte.html>.
15. **Mendoza, María A.** [En línea] [Citado el: 2 de Noviembre de 2009.]
<http://www.willydev.net/Descargas/cualmetodologia.pdf>.
16. Bitácora de un programador. [En línea] 2007. [Citado el: 3 de Noviembre de 2009.]
<http://jackopc.blogspot.com/2007/05/metodologias-rup-y-xp-procesos-de.html>.
17. Kioskea. [En línea] 2008. [Citado el: 4 de Noviembre de 2009.]
<http://es.kioskea.net/contents/cs/csiintro.php3>.
18. Mondragon. [En línea] [Citado el: 5 de Noviembre de 2009.]
http://isantolakuntza.blogspot.com/2009/04/ventajas-y-desventajas-de-la_07.html.
19. Janium. [En línea] [Citado el: 8 de Noviembre de 2009.]
<http://www.janium.com/page2/page1/page6/page7/page7.html>.
20. **Rojas, Diego.** IComparable. [En línea] 2008. [Citado el: 9 de Noviembre de 2009.]
<http://icomparable.blogspot.com/2008/11/arquitecturas-n-tier-ventajas-y.html>.
21. **Arrate, Rodrigo.** Mondragon. [En línea] 2009. [Citado el: 10 de Noviembre de 2009.]
<http://isantolakuntza.blogspot.com/2009/04/que-ventajas-y-desventajas-tiene-la.html>.
22. ProgramaciónWeb.net. [En línea] [Citado el: 11 de Noviembre de 2009.]
<http://www.programacionweb.net/articulos/articulo/?num=505>.
23. **Aguilar, José.** VariableNotFound. [En línea] 2009. [Citado el: 12 de Noviembre de 2009.]
<http://www.variablenotfound.com/2009/04/aspnet-mvc-trece-preguntas-basicas.html#p2>.

24. **Álvarez, Rubén.** DesarrolloWeb. [En línea] [Citado el: 15 de Noviembre de 2009.]
<http://www.desarrolloweb.com/articulos/534.php>.
25. Maestros del web. [En línea] [Citado el: 16 de Noviembre de 2009.]
<http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
26. Manual de CSS. [En línea] [Citado el: 17 de Noviembre de 2009.]
<http://www.manualdecss.com/manualcss/introduccion-css.html>.
27. DesarrolloWeb. [En línea] [Citado el: 18 de Noviembre de 2009.]
<http://www.desarrolloweb.com/articulos/303.php>.
28. Perl en Español. [En línea] [Citado el: 18 de Noviembre de 2009.]
http://perlenespanol.com/tutoriales/aprendiendo_perl/perl_basico_parte_1.html.
29. Ciberneta. [En línea] [Citado el: 19 de Noviembre de 2009.]
http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php.
30. Wikipedia. La Enciclopedia Libre. [En línea] [Citado el: 20 de Noviembre de 2009.]
http://es.wikipedia.org/wiki/Apache_http_server.
31. **Vera, Pablo.** Adictosalared. [En línea] 2009. [Citado el: 22 de Noviembre de 2009.]
<http://www.adictosalared.com/ Cherokee-un-servidor-web-muy-rapido-y-facil-de-configurar/>.
32. Wikipedia. La Enciclopedia Libre. [En línea] [Citado el: 23 de Noviembre de 2009.]
http://es.wikipedia.org/wiki/Mozilla_Firefox.
33. Wikipedia. La Enciclopedia Libre. [En línea] [Citado el: 23 de Noviembre de 2009.]
http://es.wikipedia.org/wiki/MySQL#Qu.C3.A9_licencia_utilizar.
34. Wikipedia. La Enciclopedia Libre. [En línea] [Citado el: 23 de Noviembre de 2009.]
<http://es.wikipedia.org/wiki/PostgreSQL>.
35. Aguilar, Vicente y Sau, Pablo. [En línea] [Citado el: 25 de Noviembre de 2009.]
<http://www.fedora-es.com/node/189>.

36. Wikipedia.La Enciclopedia Libre. [En línea] [Citado el: 27 de Noviembre de 2009.] <http://es.wikipedia.org/wiki/Framework>.
37. **Leopoldo, Carlos.** [En línea] [Citado el: 28 de Noviembre de 2009.] <http://techtastico.com/post/zend-framework-una-introduccion/>.
38. **González, Julio.** Web.Ar. [En línea] [Citado el: 29 de Noviembre de 2009.] <http://web-ar.blogspot.com/2006/10/symfony-php-framework.html>.
39. Milestone. [En línea] [Citado el: 30 de Noviembre de 2009.] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.
40. **Schmuller.** Aprendiendo UML en 24 horas. 2000.
41. Wikipedia.La Enciclopedia Libre. [En línea] [Citado el: 1 de Diciembre de 2009.] [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software)).
42. Bravo. [En línea] [Citado el: 2 de Diciembre de 2009.] <http://avbravo.blogspot.com/2008/05/soporte-php-para-netbeans.html>.
43. **Salazar, Miguel.** Manual de Herramientas CASE: Rational Rose, BoUML y JDeveloper. 2007.
44. Visual Paradigm Internacional. 10 Reasons to Choose Visual Paradigm. [En línea] 2009. [Citado el: 4 de Diciembre de 2009.] <http://www.visual-paradigm.com/aboutus/10reasons.jsp>.

Bibliografía Consultada

- **Potencier, Fabien.** *El Tutorial Jobbet.* 2009.
- **Wake, W.C.** *Extreme Programming Explored.* Addison-Wesley : s.n., 2002.
- **P. Abrahamsson, O. Ronkainen, J. Warsta.** *Agile software development methods review and analysis.* s.l. : VTT Publications, 2002.
- **Mora, Francisco.** *Estimación de costes.* Universidad de Alicante : s.n., 2002.

Glosario de Términos

Capital humano: Conjunto de conocimientos, experiencias, habilidades, sentimientos, actitudes, motivaciones, valores y capacidad para hacer, portados por los trabajadores para crear más riquezas con eficiencia. Es, además, conciencia, ética, solidaridad, espíritu de sacrificio y heroísmo.

Reclutamiento: Proceso que identifica, prepara, motiva y atrae el potencial humano para su posible selección con vista a desempeñar determinados cargos en la organización.

Selección: Capacidad del sistema para captar, seleccionar, capacitar, promover, evaluar y estimular a las personas de acuerdo a los valores de la organización laboral.

Cargo: Los que aparecen recogidos en los calificadores comunes, de rama o actividad y propios de organismos, pertenecientes a las diferentes categorías ocupacionales, así como en resoluciones. En ellos se definen la denominación, contenido de trabajo y los requisitos para ocuparlos y expresan el empleo u oficio que desempeñan, los trabajadores en la organización.

Categoría ocupacional: Clasificación de los diferentes cargos, de acuerdo con las funciones y tareas que desarrollan los trabajadores, tomando en cuenta sus características disímiles, la complejidad y su vinculación con la producción y la prestación de servicios. Se agrupan en las categorías de operarios, trabajadores de servicios, trabajadores administrativos, técnicos y dirigentes.