



INSTITUTO SUPERIOR MINERO METALÚRGICO

“Dr. Antonio Núñez Jiménez”

Facultad de Geología -Minas

Trabajo de Diploma

Para optar por el Título de Ingeniero Informático

INTRANET CORPORATIVA EMPLNI MÓDULO GESTIÓN DE CAPACITACIÓN

Autor: Leonart Urgellés Olivero

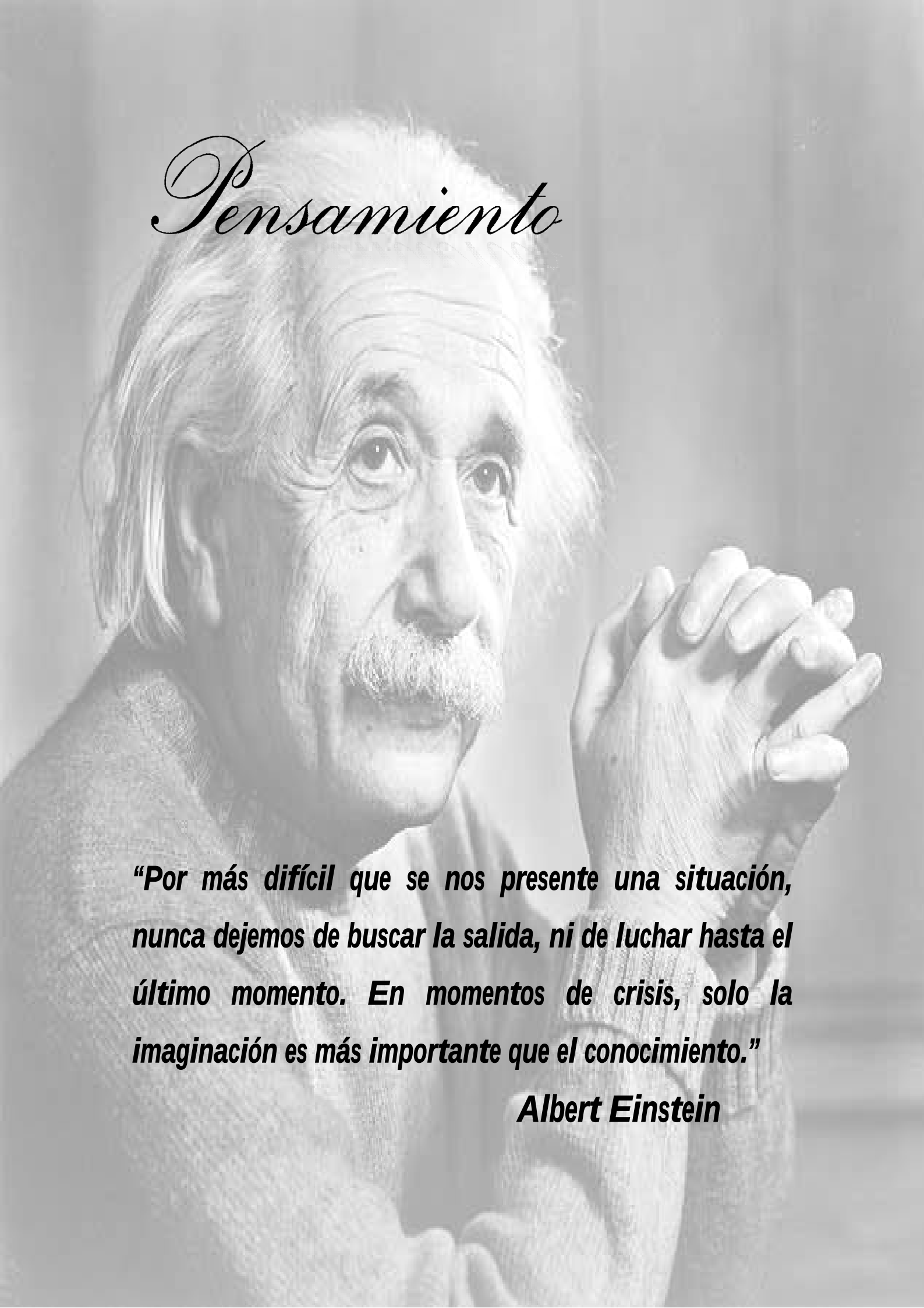
Tutores: Ing. Edgar Núñez Torres

Ing. Jorge Mario Basulto Aguilera

Consultante: Lic. Eudis Antonio Perdomo Orellana

Moa, Holguín, Cuba

2012



Pensamiento

“Por más difícil que se nos presente una situación, nunca dejemos de buscar la salida, ni de luchar hasta el último momento. En momentos de crisis, solo la imaginación es más importante que el conocimiento.”

Albert Einstein

Agradecimientos

Quiero
expresar mi agradecimiento a la Revolución Cubana por permitirme
formarme profesionalmente.

*A mi familia, especialmente a mis padres los mejores de mundo por
brindarme su amor apoyo en todo momento.*

*A mis hermanos que siempre estuvieron pendientes de mí dándome
una dosis de ánimo cuando me hizo falta.*

*A mi cosi Lianet por estar a mi lado todo el tiempo, por confiar en
mí y brindarme lo mejor de su energía.*

A sus padres por darme un voto de confianza.

*A mis compañeros de estudio que me aguantaron estos 5 años en
especial al Tino y a Jesús que son partícipes de muchas de mis
locuras.*

A mis amigos.

*A mis tutores Edgar, Mario y Eudis por brindarme una mano guía
para enseñarme el camino a transitar.*

*Al colectivo de profesores del ISMM y a los trabajadores de la
EMPLENI.*

A todos mi eterno agradecimiento...

Leonart Urgelles Olivero

Dedicatoria

A mis Padres, A mi familia a mi novia y a mis amigos...

Leonart Urgelles Olivero

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Empresa Empleadora del Níquel para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los... días del mes... del año 2012.

Leonart Urgellés Olivero

Nombre completo del autor

Ing. Edgar Núñez Torres

Nombre completo del primer tutor

Ing. Jorge Mario Basulto Aguilera

Nombre completo del segundo tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El trabajo de diploma, titulado **INTRANET CORPORATIVA EMPLeni. MÓDULO GESTIÓN DE CAPACITACIÓN**, fue realizado en nuestra entidad Empresa Empleadora del Níquel "EMPLeni". Se considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

☐ Totalmente

☐ Parcialmente en un ____%.

Los resultados de este informe técnico le reportan a esta entidad los beneficios siguientes:

Y para que así conste, se firma la presente a los__días del mes de_____del año 2012

Nombre del representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: *"INTRANET CORPORATIVA EMPLNI. MÓDULO GESTIÓN DE CAPACITACIÓN".*

Autor: Leonart Urgellés Olivero

Ing. Edgar Núñez Torres.

Ing. Jorge Mario Basulto Aguilera

Fecha: _____

RESUMEN

En los momentos actuales las empresas de nuestro país desarrollan diferentes actividades en correspondencia con su utilidad social, tareas que son llevadas a cabo por trabajadores, pero qué pasaría si un trabajador no se encuentra al tanto con la actividad que realiza o que debe realizar. Una forma muy factible de instruirlo es mediante la capacitación, método mediante el cual se brinda enseñanza-aprendizaje proporcionada fuera del sistema escolar establecido. La función principal de este trabajo es precisamente la realización de una aplicación informática capaz de optimizar la gestión de información referente al proceso de capacitación de recursos humanos en la Empresa Empleadora del Níquel en Moa, proceso que en la actualidad se realiza de manera descentralizada debido a que la herramienta informática con que se cuenta no permite una adecuada distribución y actualización de sus funcionalidades lo que no hace óptimo el manejo de datos necesarios para la realización de las actividades de capacitación.

La aplicación fue desarrollada mediante las fases de la metodología xp y con el apoyo de tecnologías y herramientas como el lenguaje de programación php, el sistema gestor de base de datos MySQL y el servidor Web apache.

ABSTRACT

At present, country's companies develop different activities according to its social usefulness instead to. These activities are carried out by workers, but what if a worker is not aware of the work performed or to be performed. A very feasible way to instruct workers is through training, the method by means of which teaching and learning is provided outside the school system already established. The main function of this work is precisely the fulfillment of a Web application able to optimize the management of information concerning the training process of human resources in the Employer Nickel Company (EMPLENI) in Moa. A process that currently takes place on a decentralized way because the computer application does not allow a proper distribution and updating of its features, so that the management of needed data is not the best to carry out training activities.

This application was developed using extreme programming methodology phases and for its development were used PHP as programming language, MySQL as database management system and Apache as web server.

INDICE

DECLARACIÓN DE AUTORÍA	IV
OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA.....	V
OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA.....	VI
RESUMEN	VII
ABSTRACT	VIII
INDICE.....	IX
INTRODUCCIÓN	1
CAPITULO I – FUNDAMENTACIÓN TEÓRICA.	6
1.1 INTRODUCCIÓN	6
1.2 ESTADO DEL ARTE.....	6
1.2.1 Conceptualización y características de la capacitación.....	6
1.2.2 Antecedentes Históricos de la Capacitación.....	7
1.2.3 La gestión de la capacitación en las empresas.....	9
1.3 SISTEMAS INFORMÁTICOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN	13
1.4 ¿QUÉ ES UNA APLICACIÓN WEB? ¿POR QUÉ LA UTILIZAMOS?	14
1.5 TECNOLOGÍAS Y HERRAMIENTAS A UTILIZAR.	18
1.5.1 Lenguajes de programación	18
1.5.2 Sistemas gestores de bases de datos.....	20
1.5.3 Servidores para aplicaciones WEB.....	22
1.5.4 Paquete de tecnología Web.....	24
1.5.5 Dreamweaver como un entorno de Desarrollo.....	25
1.5.6 EMS SQL Manager para MySQL.....	27
1.6 METODOLOGÍAS EXISTENTES	28
1.6.1 Metodologías Tradicionales	28
1.6.2 Metodologías Ágiles.....	30
1.7 ESTILO ARQUITECTÓNICO:	38
1.8 CONCLUSIONES.....	41
CAPITULO II - PLANIFICACIÓN Y DISEÑO.....	42
2.1 INTRODUCCIÓN.....	42
2.2. PERSONAL RELACIONADO CON EL SISTEMA	42
2.3 FUNCIONALIDADES	43
2.3.1 Requisitos funcionales.....	43
2.3.2 Requisitos no funcionales.....	44
2.4 HISTORIAS DE USUARIO.....	46
2.5 PLANIFICACIÓN DE ENTREGAS.....	47
2.5.1 Estimación de esfuerzo por historias de usuario.....	48
2.5.2 Planificación de iteraciones.....	48
2.5.3 Plan de duración de las iteraciones.....	50
2.6 TARJETAS CLASES-RESPONSABILIDADES-COLABORACIÓN (CRC).....	50
2.7 DESCRIPCIÓN DE LAS INTERFACES DE USUARIO.	51
2.8 CONCLUSIONES.....	53
CAPITULO III – DESARROLLO Y PRUEBAS.....	54

3.1 INTRODUCCIÓN.....	54
3.2 DESARROLLO DE LAS ITERACIONES.....	54
3.2.1 Tareas por historias de usuario.....	54
3.3 MODELADO DE DATOS.....	56
3.4 PRUEBAS.....	57
3.4.1 Pruebas de aceptación.....	58
3.5 CONCLUSIONES.....	59
CAPITULO IV – ESTUDIO DE FACTIBILIDAD	60
4.1 INTRODUCCIÓN.....	60
4.2 ESTUDIO DE FACTIBILIDAD	60
4.2.1 Factibilidad técnica.....	60
4.2.2 Factibilidad económica.....	62
4.2.2.1 Efectos económicos.....	62
4.2.2.2 Fichas de costo.....	64
4.2.3 Factibilidad operativa.....	67
4.3 CONCLUSIONES.....	68
CONCLUSIONES GENERALES.....	69
RECOMENDACIONES.....	70
BIBLIOGRAFÍAS	71
ANEXO 1: HISTORIAS DE USUARIO.....	74
ANEXO 2: TARJETAS CRC.....	78
ANEXO 3: INTERFACES DE USUARIO	82
ANEXO 4: TARJETAS DE TAREAS.....	86
ANEXO 5: PRUEBAS DE ACEPTACION	102

INTRODUCCIÓN

En la actualidad no se concibe un mundo en el cual los procesos desarrollados por personas no evolucionen en su forma de realización. Parte de esta inevitable evolución es la informatización de eventos reales, que aplicada a conceptos de ahorro de materiales y rapidez en los servicios es una de las más factibles. La influencia que esta tiene para el mundo pone en evidencia la importancia de la relación hombre-máquina como eslabón fundamental en el desarrollo económico y social de los países, logrando integrarse a casi todas las esferas de la vida, acrecentando la dependencia y la necesidad en muchas ocasiones de automatizar los procesos cotidianos con el objetivo de viabilizar la gestión de la información.

El alcance que tienen hoy en día las Tecnologías de la Información y las Comunicaciones en Cuba, se hace evidente en la incorporación de la informática a las esferas productivas que es ya una necesidad, se hace evidente las formas de utilización de estas tecnologías de la información en las diferentes esferas del modelo económico que se desenvuelve, trayendo consigo en muchas ocasiones el ahorro de tiempo, recursos y facilidad de acceso a tecnologías sin importar la distancia de comunicación.

Contribuyendo al proceso de informatización en nuestro país se forman profesionales en las distintas universidades del sistema educacional cubano capaces de resolver las más disímiles tareas, estos desarrollan día a día múltiples aplicaciones informáticas orientadas a resolver las necesidades que tenemos, las herramientas desarrolladas son aplicadas a empresas y entidades que necesitan de sus servicios.

En nuestro municipio la producción de software brinda soluciones a muchos problemas relacionados con las industrias locales. Teniendo en cuenta las ventajas de las tecnologías de la información y las comunicaciones, la Empresa Empleadora del Níquel (EMPLENI) se ha propuesto informatizar una gran parte

de los procesos desarrollados en la misma. Dentro de estos los de capacitación se encuentran en una fase desarrollada pero no óptima, debido a que la información con la que se trabaja se desarrolla sobre un software que no cumple en su totalidad con las propiedades y necesidades de la empresa, debido a lo tedioso del trabajo con el mismo, la falta de funcionalidades y acceso. Problemas que se pudieran solucionar si existiera una aplicación capaz de gestionar y optimizar los procesos de capacitación con mayor fiabilidad.

Haciendo un análisis profundo del problema antes expuesto se define como **problema científico**: ¿Cómo optimizar la gestión de la información en el proceso de capacitación en la Empresa Empleadora del Níquel mediante el uso de las Tecnologías de la Información y las Comunicaciones?

En busca de una solución para el problema antes planteado definimos como **objeto de estudio** los Sistemas informatizados para la gestión del proceso de capacitación, el **campo de acción** se enmarca en la informatización de la gestión de información del proceso de capacitación en la Empresa Empleadora del Níquel.

Para dar solución al problema antes mencionado se plantea como **objetivo general**:

Desarrollar una aplicación informática para optimizar la gestión de información del proceso de capacitación en la EMPLNI.

Planteando como **idea a defender** que si se desarrolla una aplicación informática para la gestión de información del proceso de capacitación en la EMPLNI, se optimizará el manejo de la misma para la Dirección de Recursos Humanos de la empresa.

En correspondencia con esta propuesta se presentan los siguientes **objetivos específicos**:

1. Establecer el estado del arte sobre la información disponible relacionada con la capacitación.
2. Caracterizar los sistemas informáticos existentes vinculados al objeto de estudio.
3. Estudiar las tecnologías y herramientas a utilizar para el diseño e implementación de la aplicación.
4. Implementar una aplicación informática que permitirá llevar a cabo la gestión de información del proceso de capacitación en la empresa.

Para dar cumplimiento a los objetivos específicos se plantearon las siguientes **tareas específicas:**

1. Análisis bibliográfico de las literaturas científicas relacionadas con el proceso de capacitación.
2. Análisis de las ventajas y desventajas de los sistemas informáticos existentes.
3. Selección del lenguaje de programación y de las herramientas para implementar la aplicación informática.
4. Estudio y selección de las metodologías para el desarrollo del software.
5. Diseño e implementación de la aplicación informática para la gestión de información de capacitación.
6. Realización de pruebas al sistema informático creado.

Entre los métodos utilizados para el desarrollo de estas tareas científicas podemos mencionar:

Métodos teóricos:

Histórico-lógico: para la búsqueda de antecedentes del software, las herramientas utilizadas, así como la forma en que se desarrollaba el proceso de capacitación la EMPLNI.

Análisis y síntesis: Se utilizó en los fundamentos teóricos, en el procesamiento de la información y en la descomposición de cada uno de los requerimientos del sistema.

Revisión y análisis de documentos: Fue usado en la fundamentación de los requisitos del sistema y en la comprensión de los procesos desarrollados así como en la elaboración los fundamentos teóricos que se relacionan con el campo de acción.

Métodos empíricos:

Observación: Fue empleada en la caracterización de la ejecución de las actividades de capacitación vigentes en la empresa, así como para llevar a cabo el desarrollo y evaluación del resultado de la investigación.

Entrevistas: Se empleó en la determinación de los requerimientos funcionales del sistema que se va implementar. Se llevó a cabo un diálogo con personas expertas en la materia.

El trabajo consta de introducción, cuatro capítulos, conclusiones generales, recomendaciones, bibliografías y anexos:

Introducción: en esta se describen la problemática existente, el problema científico, objeto de estudio, campo de acción, objetivo general, idea a defender, objetivos específicos y tareas específicas.

Capítulo I " Fundamentación Teórica: En este se brinda una descripción de los conceptos relacionados al objeto de estudio, el objetivo general y las tareas, así como un análisis de los sistemas existentes vinculados al campo de acción, también se abordan temas específicos relacionados con la metodología y las herramientas a utilizar para el desarrollo de la aplicación.

Capítulo II " Planificación y Diseño: Se pone en práctica la utilización de la metodología abordada en el capítulo 1 y detalles específicos en cada una de sus fases, la elaboración de las tarjetas CRC además de los requisitos de sistema para el desarrollo del proyecto y propuestas de interfaz de usuario.

Capítulo III " Desarrollo y Pruebas: Se presentan los principales métodos y definiciones dentro de la implementación de los flujos de trabajo. Se describen además las pruebas realizadas y sus resultados.

Capítulo IV " Estudio de factibilidad: Se hace un análisis de los esfuerzos necesarios para la creación del sistema, su factibilidad y un gráfico de comparación de ventajas de la realización del proceso de la forma anterior y de la forma optimizada.

Conclusiones generales y recomendaciones: las primeras fueron elaboradas teniendo en cuenta los objetivos y tareas propuestas, las segundas con vista a mejoras futuras del producto informático.

Bibliografías y anexos: Se muestran las bibliografías utilizadas para la elaboración del producto y anexos que respaldan la metodología seleccionada.

CAPITULO I – FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción

En el presente capítulo se tratan temas de vital importancia para el desarrollo del software como son el concepto de capacitación y sus características principales, así como las herramientas a utilizar para la implementación del producto y el análisis de los Gestores de Base de Datos, lenguajes de programación, metodologías y patrones arquitectónicos existentes para determinar cuáles son más factibles atendiendo a las ventajas y desventajas de cada uno de ellos.

1.2 Estado del arte

1.2.1 Conceptualización y características de la capacitación.

Definición de capacitación:

Para poder tener un concepto claro sobre la capacitación, es necesario diferenciarlo del entrenamiento y el adiestramiento. El entrenamiento es la preparación que se sigue para desempeñar una función.

Mientras que el adiestramiento es el proceso mediante el cual se estimula al trabajador a incrementar sus conocimientos, destreza y habilidad. En cambio, capacitación es la adquisición de conocimientos técnicos, teóricos y prácticos que van a contribuir al desarrollo de individuos en el desempeño de una actividad. Se puede señalar, entonces, que el concepto capacitación es mucho más abarcador. (<http://rrhh-web.com/capacitacion.html>)

La capacitación es una herramienta fundamental para la Administración de Recursos Humanos, es un proceso planificado, sistemático y organizado que busca modificar, mejorar y ampliar los conocimientos, habilidades y actitudes del personal nuevo o actual, como consecuencia de su natural proceso de cambio, crecimiento y adaptación a nuevas circunstancias internas y externas.

La capacitación mejora los niveles de desempeño y es considerada como un factor de competitividad en el mercado actual.

A continuación se presentan algunos conceptos sobre Capacitación de personal:

La capacitación consiste en proporcionar a los empleados, nuevos o actuales, las habilidades necesarias para desempeñar su trabajo. Proceso de enseñanza de las aptitudes básicas que los nuevos empleados necesitan para realizar su trabajo.”(Gary Dessler 1998)

La capacitación se considera como un proceso a corto plazo, en que se utiliza un procedimiento planeado, sistemático y organizado, que comprende un conjunto de acciones educativas y administrativas orientadas al cambio y mejoramiento de conocimientos, habilidades y actitudes del personal, a fin de propiciar mejores niveles de desempeño compatibles con las exigencias del puesto que desempeña, y por lo tanto posibilita su desarrollo personal, así como la eficacia, eficiencia y efectividad empresarial a la cual sirve. Es un proceso educativo a corto plazo, aplicado de manera sistemática y organizada, mediante el cual las personas aprenden conocimientos, actitudes y habilidades, en función de objetivos definidos. El entrenamiento implica la transmisión de conocimientos específicos relativos al trabajo, actitudes frente a aspectos de la organización, de la tarea y del ambiente, y desarrollo de habilidades. Cualquier tarea, ya sea compleja o sencilla, implica necesariamente estos tres aspectos.”(www.wikilearning.com/.../capacitacion)

1.2.2 Antecedentes Históricos de la Capacitación

Si queremos comprender el surgimiento y la necesidad de la capacitación en la sociedad debemos remontarnos a la manera en que trabajaban las personas en los tiempos remotos y como se dieron los primeros indicios de entrenamiento desde las primeras construcciones del hombre. Estas nos

permiten concluir que para su construcción intervinieron personas con diferentes oficios, como pintores, escultores, carpinteros y arquitectos.

En la antigüedad, la alfabetización se limitaba a ciertos sectores sociales y la única manera que había de comunicar los conocimientos era mediante la transmisión verbal de generación en generación, indicando y supervisando la ejecución de los oficios o actividades y de esta forma es como se entrenaba a familias completas y se especializaban en algún oficio o actividad (Grados, 2004)

Los gremios constituyen la primera forma del concepto de una empresa y estas a su vez dan origen a las agrupaciones de trabajadores. Estos se crearon para proteger los intereses de los trabajadores, hoy en día se conocen como sindicatos y con ellos surgieron reglamentaciones como la cantidad y la calidad de trabajo que debía ejecutar el trabajador tomando en cuenta el sistema de remuneración.(catarina.udlap.mx/u_dl_a/tales/documentos/lhr/.../capitulo2.pdf)

Conforme al paso del tiempo los mercados se fueron expandiendo, por lo tanto se requirió más maquinaria y materiales, y esto a su vez propició una mayor inversión por parte de los maestros. Con esto nació la necesidad de parte de los trabajadores de instalar talleres propios y esto hizo que hubiera trabajadores especializados. Esto también fue propiciado por los maestros, que al ver establecido nuevos talleres hicieron los suyos mas exclusivos y por lo tanto demandaban mayor habilidad en la mano de obra que contrataban (Grados, 2004).

Cabe mencionar que la capacitación se constituyó más que como un hecho educativo, como una fórmula para controlar los secretos de los diferentes oficios con el propósito de proteger intereses económicos y estatus social de artesanos y comerciantes.

Con la Revolución Industrial, la capacitación se transformó con la incorporación de objetivos y métodos, es decir, se pedía la participación de las personas para

que pudieran trabajar en una sola actividad, teniendo a su cargo una parte del proceso de fabricación a realizar. Lo único que se hacía era entrenarlas en una tarea del proceso y no en todas las actividades. Esto fue lo que provocó la Revolución Industrial junto con los problemas de los obreros y la explotación de los mismos. (catarina.udlap.mx/u_dl_a/tales/documentos/lhr/.../capitulo2.pdf)

A medida que los empresarios fueron adquiriendo mayor crecimiento y desarrollo para sus empresas fueron entrenando a los trabajadores no solo en las actividades que realizaban sino en diversas actividades y así es como empiezan a participar más en los procesos que se llevan a cabo(Grados, 2004).

1.2.3 La gestión de la capacitación en las empresas.

Teniendo en cuenta que la motivadora principal de las empresas es la capacitación. El colaborador que recibe capacitación siente que la empresa lo estima y por lo tanto, le está asignando un salario espiritual y considera que están invirtiendo en su talento para mejorar su rendimiento, la calidad de su trabajo, elevar su productividad y consecuentemente, piensa que puede estar próximo a un ascenso.

Si bien es cierto que el aumento del salario económico es importante para mejorar la calidad de vida, también es cierto que, pasado cierto período, la nueva remuneración se diluye en satisfacer ciertas necesidades y nuevamente, se requiere nuevo aumento; en cambio, el salario espiritual permite mejorar la calidad humana del hombre, coadyuva a la felicidad de su hogar.

El desarrollo de los recursos humanos es central ante el reto tan importante que las empresas enfrentan en este mundo globalizado y competitivo.

La capacitación y desarrollo del recurso humano, es una estrategia empresarial importante que deben acompañar a los demás esfuerzos de cambio que las organizaciones lleven adelante.

Mediante esta estrategia los colaboradores aprenden cosas nuevas, crecen individualmente, establecen relaciones con otros individuos, coordinan el trabajo a realizar, se ponen de acuerdo para introducir mejoras, etc, en otras palabras les convienen tanto al colaborador como a la empresa, por cuánto los colaboradores satisfacen sus propias necesidades y por otra parte ayudan a las organizaciones alcanzar sus metas; como podrá apreciarse la capacitación y desarrollo comienza con una inversión que las empresas deberán poner atención, e invertir más para lograr con eficiencia y rentabilidad mejores logros.

La capacitación ha demostrado ser un medio muy eficaz para hacer productivas a las personas, su eficacia se ha demostrado en más del 80% de todos los programas de capacitación. No obstante, los directivos no deben dar por hecho que exista una relación causal entre el conocimiento impartido o las destrezas enseñadas y el aumento significativo de la productividad.

La capacitación no sólo es buena para la empresa, sino que también es un buen negocio. Ya a mediados de los años 50, la mitad de las compañías importantes contaban con algún tipo de programa de capacitación. A finales de los años 70, el 75% de las empresas importantes tenían departamentos independientes dedicados exclusivamente a ofrecer una gama completa de programas de capacitación

La formación y el perfeccionamiento son áreas en las que ha aumentado el interés en los últimos años. Decidir, diseñar y poner en marcha programas de formación y perfeccionamiento de los empleados, con el objetivo de mejorar sus capacidades, aumentar su rendimiento y hacerlos crecer es una cuestión por la que cada vez hay una mayor preocupación importante por la capacitación.

Los cambios rápidos que se producen en las tecnologías y la necesidad de disponer de una fuerza laboral que sea continuamente capaz de llevar a cabo

nuevas tareas, supone un importante reto al que tiene que hacer frente los departamentos de recursos humanos.

En conclusión la capacitación es importante, porque permite:

- Consolidación en la integración de los miembros de la organización.
- Mayor identificación con la cultura organizacional.
- Disposición desinteresada por el logro de la misión empresarial.
- Entrega total de esfuerzo por llegar a cumplir con las tareas y actividades.
- Mayor retorno de la inversión.
- Alta productividad.
- Promueve la creatividad, innovación y disposición para el trabajo.
- Mejora el desempeño de los colaboradores.
- Desarrollo de una mejor comunicación entre los miembros de una organización.
- Reducción de costos.
- Aumento de la armonía, el trabajo en equipo y por ende de la cooperación y coordinación.
- Obtener información de fuente confiable, como son los colaboradores.

(www.mitecnologico.com/Main/ConceptoImportanciaCapacitacion)

La capacitación en Cuba

En Cuba los sistemas de capacitación están normados por el Ministerio de Trabajo y Seguridad Social, que regula la capacitación profesional de los trabajadores como un proceso permanente y planificado, concebido como una inversión para el desarrollo que llevan a cabo las organizaciones empresariales con el objetivo de que sus trabajadores adquieran y perfeccionen su competencia laboral, de modo que estén aptos para responder a las exigencias constantemente cambiantes del proceso productivo o de prestación de

servicios del que forman parte. La Resolución No. 29/06 define la capacitación como el conjunto de acciones de preparación que desarrollan las entidades laborales dirigidas a alcanzar habilidades, destrezas y valores en los trabajadores que les permita desempeñar puestos de trabajos de perfil amplio.

En nuestro país, el Estado asume el rol de financiador, regulador y supervisor de la calidad de la capacitación y de las adecuadas condiciones de competencia. Dejando el rol central a las propias empresas demandantes de capacitación, quienes facilitan la inserción posterior en el empleo para el que se está capacitando el individuo. La entidad capacitadora integra a su propuesta la práctica laboral, en estrecha vinculación con las entidades empleadoras. Muchas empresas asumen el contrato de trabajo en período de aprendizaje, tal y como lo regula el Código del Trabajo; pero lo ideal sería que las empresas contaran con sus entidades capacitadoras que cumplan ciertos requisitos mínimos y sean de reconocimiento institucional y estatal, para que se diversifique la oferta de capacitación, tanto por parte de empresas como de otras organizaciones y, se asegure que la formación se realice en ocupaciones para las cuales exista demanda explícita por parte de las empresas, asegurando la actualización y adaptación a los nuevos perfiles ocupacionales que pueden ir surgiendo con los cambios en las tecnologías y las actividades económicas, políticas y sociales.

(http://www.ecured.cu/Capacitación_en_América_Latina_y_Cuba)

Gestión de la capacitación en la Empresa Empleadora del Níquel (EMPLENI)

Para la realización de proyectos es necesaria la comprensión de sus procesos y su flujo de trabajo, motivo por el cual es primordial el establecimiento de los eventos sobre los que se maneja la información.

Para el desarrollo de este proyecto fue necesario realizar un análisis de carácter informativo en el departamento de recursos humanos de la EMPL

y dentro de este del proceso de capacitación, el cual se desarrolla de la siguiente manera:

El proceso inicia cuando la alta dirección determina las necesidades de capacitación y desarrollo para los trabajadores, mediante un proceso continuo e ininterrumpido, en correspondencia con los cargos que ocupan.

Los máximos responsables de la elaboración del diagnóstico o determinación de las necesidades de capacitación son los jefes inmediatos de cada área. Para el diagnóstico de la determinación de las necesidades de capacitación (DNC) se debe partir del Perfil de Competencias determinado para cada cargo y trabajador, según PG-DRH-009 “Procedimiento Identificación y Validación de Competencias”. A cada competencia del cargo se le realizará una evaluación cruzada de:

- La evaluación del jefe inmediato del trabajador
- Evaluación de un experto.
- Auto evaluación del trabajador

La DNC de cada trabajador tiene como resultado la identificación de las brechas que este presenta entre las competencias laborales requeridas para el cargo y las que posee realmente. Como resultado de este proceso cada jefe de área emitirá la propuesta de necesidades de capacitación de cada trabajador subordinado a él, quedando reflejado en el registro “Propuestas de las Necesidades de Capacitación por orden de prioridades”. En correspondencia con las necesidades de cada trabajador se determinan las acciones a realizar para su capacitación.

1.3 Sistemas informáticos existentes vinculados al campo de acción

Qtraining (Software para Evaluación de Desempeño y Control de Capacitación)

Es un software propietario que posee una poderosa pero sencilla herramienta para evaluar el desempeño del personal, detectar necesidades de capacitación y administrar los cursos de capacitación de la empresa. Permite evaluar el impacto de las capacitaciones llevadas a cabo y relaciona los objetivos de desempeño con los cursos de capacitación.

CAP (Sistema de Gestión de la capacitación en la EMPLENI)

Sistema informático creado con anterioridad en la EMPLENI con el objetivo de informatizar la gestión de información de capacitación en la entidad, es una aplicación de escritorio desarrollada sobre una base datos en SQL Server con interfaz en Access que brinda a los usuarios facilidades con la manipulación de información de capacitación.

¿Por qué no utilizarlos?

Los software antes mencionados tienen una amplia gama de aplicabilidad, sin embargo, estos no se adaptan a las necesidades que se requieren en la empresa para resolver de forma objetiva y factible el problema existente.

El primero trabaja de manera general con los procesos de capacitación, pero a parte de su costo, no tiene incorporadas las funcionalidades específicas que necesita el centro para resolver su problema, haciéndose muy difícil su adaptación al centro.

El segundo que está en explotación, presenta dificultades en cuanto a la gestión de la información, a causa de la falta de funcionalidades y a lo difícil de su distribución, mantenimiento y actualización, lo que trae consigo atrasos en los planes de capacitación y su difícil vinculación a la intranet de la entidad.

1.4 ¿Qué es una aplicación Web? ¿Por qué la utilizamos?

Una aplicación Web es un sistema informático que los usuarios usan accediendo a un servidor Web a través de los protocolos de Internet. Las

aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero. La facilidad para actualizar y mantener las aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su creciente popularidad.

Una aplicación Web está comúnmente estructurada como una aplicación en tres-capas. En su forma más común el navegador es la primera capa, un motor usando alguna tecnología de Web dinámica (Ej.: CGI, PHP, ASP, Java Servlets o Pearl) es la capa del medio, y algún tipo de gestor de bases de datos como última capa. (Subiros 2009).

En tiempos recientes se ha usado la estrategia de generalizar esta arquitectura mediante la adición de piezas de hardware que permitan balancear la carga de los servidores Web y de las aplicaciones.

Las Aplicaciones Web son de un desarrollo poco costoso, sencillo y rápido.

Presenta acceso ubicuo, sin necesidad de distribución e idealmente, con pocos requerimientos técnicos. Con datos centralizados y fácil integración de datos múltiples fuentes. (Subiros, 2009).

Por los aspectos antes expuestos se ha decidido a usar la tecnología Web para el desarrollo del sistema.

Ventajas de una WEB: (Graham, 2001)

- **Compatibilidad multiplataforma:** Las aplicaciones Web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.
- **Actualización:** Las aplicaciones basadas en Web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o

interferir con sus hábitos de trabajo, pues no se hace necesario iniciar nuevas descargas y/o procedimientos de instalación (algunas veces imposible cuando usted está trabajando dentro de grandes organizaciones).

- **Inmediatez de acceso:** Las aplicaciones basadas en Web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online y están listas para trabajar sin importar cuál es su configuración o su hardware.
- **Menos requerimientos de memoria:** Las aplicaciones basadas en Web tienen menos demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, esas aplicaciones basadas en Web usan en muchos casos la memoria de las computadoras donde ellas corren, dejando más espacio para correr múltiples aplicaciones sin incurrir en frustrantes deterioros en el rendimiento.
- **Menos Bugs:** Las aplicaciones basadas en Web son menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en Web, todos utilizan la misma versión, y todos los bugs pueden ser corregidos tan pronto como son descubiertos.
- **Precio:** Las aplicaciones basadas en Web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes descargables y no totalmente gratuitas, mientras que ofrecen componentes adicionales y servicios Premium como una opción.
- **Los datos también van online:** El hecho de que el manejo de los datos sea realizado de forma remota libra al usuario de la responsabilidad en la protección de los mismos y al mismo tiempo logra que los recursos sean accesibles en cualquier momento.

- **Múltiples usuarios concurrentes:** Las aplicaciones basadas en Web pueden ser utilizadas por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta.
- **Los datos son más seguros:** Si bien la ruptura de discos no va a desaparecer, es probable que los usuarios escuchen mucho menos del tema. A medida que las compañías se hagan cargo del almacenamiento de los datos del usuario, granjas de almacenamiento de datos redundantes, altamente fiables, los usuarios van a tener mucho menos riesgo de perder sus datos debido a una ruptura de disco impredecible o a un virus de la computadora. Las compañías que proveen aplicaciones basadas en Web van a brindar amplios servicios de resguardo de datos ya sea como una parte integral del servicio básico o como una opción paga.
- **Desarrollar aplicaciones en el lenguaje que usted quiera:** Una vez que las aplicaciones han sido separadas de computadoras locales y sistemas operativos específicos, pueden también ser escritas en prácticamente cualquier lenguaje de programación. Debido a que las aplicaciones Web son esencialmente una colección de programas más que un simple programa, ellas podrían ser escritas en cualquier lenguaje de programación existente.

Desventajas WEB: (Masternewmedia.org, 2005)

- Acceso limitado, la necesidad de conexión permanente y rápida a Internet hacen que el acceso a estas aplicaciones no esté al alcance de todos.
- La interactividad no se produce en tiempo real, en las aplicaciones Web cada acción del usuario conlleva un tiempo de espera hasta que se obtiene la reacción del sistema.

- Elementos de interacción muy limitados. En comparación con el software de escritorio, las posibilidades de interacción con el usuario que ofrecen las aplicaciones Web (mediante formularios principalmente) son muy escasas.
- Diferencias de presentación entre plataformas y navegadores. La falta de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones.

1.5 Tecnologías y Herramientas a utilizar.

1.5.1 Lenguajes de programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. Pueden ser agrupados en dos grupos:

Del lado del cliente:

JavaScript: este es un lenguaje interpretado por lo que no requiere compilación. Fue creado por la empresa Netscape Communication. Es similar al Java, aunque no es un lenguaje orientado a objetos, no dispone de ninguno de los privilegios de la programación orientada a objetos (POO). La mayoría de los navegadores en sus últimas versiones interpretan JavaScript. El código de JavaScript puede ser integrado a nuestras páginas Web. (<http://www.ecured.cu/>)

Del lado del Servidor

PHP

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser

utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK34.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools o Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994. (Achour, 2005)

Justificación del lenguaje de programación a utilizar.

¿Porque PHP?

Ventajas de PHP

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones. (<http://www.ecured.cu/>)

1.5.2 Sistemas gestores de bases de datos

Un Sistema Gestor o Manejador de Bases de Datos (SGBD) es un conjunto de programas que permite a los usuarios crear y mantener una BD, por lo tanto, el SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Pueden ser de propósito general o específico.(CAVSI 2004)

SQLServer

SQL Server es un sistema de gestión de bases de datos relacionales (SGDBR o RDBMS: Relational Database Management System) diseñado para trabajar con grandes cantidades de información y la capacidad de cumplir con los requerimientos de proceso de información para aplicaciones comerciales y sitios Web. SQL Server ofrece el soporte de información para las tradicionales aplicaciones Cliente/Servidor, las cuales están conformadas por una interfaz a través de la cual los clientes acceden a los datos por medio de una LAN.

La hoy emergente plataforma NET exige un gran porcentaje de distribución de recursos, desconexión a los servidores de datos y un entorno descentralizado, para ello sus clientes deben ser livianos, tales como los navegadores de Internet los cuales accederán a los datos por medio de servicios como el Internet Information Services(IIS).

(http://es.wikipedia.org/wiki/Microsoft_SQL_Server.)

MySQL - es un sistema de gestión de base de datos relacional, multi-hilo y multiusuario, con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. MySQL AB pertenece a Sun Microsystems desde enero de 2008. Por un lado lo ofrece bajo la GNU GPL, pero, entidades que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso. Está desarrollado en su mayor parte en ANSIC.

Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius.

MySQL funciona sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows. También existe MySQL para OpenVMS. (es.wikipedia.org/wiki/MySQL)

Justificación del (SGBD) a utilizar.

¿Por qué MySQL?

MySQL cuenta con muchas ventajas, entre las que se destacan:

- **Alto rendimiento:** es muy rápido con respecto a otros gestores de bases de datos.
- **Bajo coste:** está disponible de manera gratuita, bajo una licencia de código abierto, o por un precio reducido en forma de licencia comercial si resultara necesario para su aplicación.
- **Facilidad de configuración y aprendizaje:** Las bases de datos más modernas utilizan SQL. Si ha utilizado otros RDBMS, no debería tener problemas para adaptarse a este sistema. MySQL resulta además más sencillo de configurar que otros productos similares.

- **Portabilidad:** se puede utilizar en una gran cantidad de sistemas Unix diferentes así como bajo Microsoft Windows.
- **Accesibilidad a código fuente:** Como en el caso de PHP, puede obtener y modificar el código fuente de MySQL.

1.5.3 Servidores para aplicaciones WEB

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. (Vegas, 2002)

Zope

Es un servidor de aplicaciones web de código abierto escrito en el lenguaje de programación Python. Puede ser manejado casi totalmente usando una interfaz de usuario basada en páginas Web.

Un sitio web de Zope está compuesto de objetos en lugar de archivos, como es usual con la mayoría de los otros sistemas de servidores web. Las ventajas de usar objetos en lugar de archivos son:

- Combinan el comportamiento y los datos en una forma más natural que los archivos de texto plano.
- Alientan el uso de componentes estándares que se ocupan de una parte particular de las que forman una aplicación Web, permitiendo flexibilidad y buena descomposición.
- Posibilitan procesos automáticos de gestión de información.

Lo más característico de Zope es su base de datos orientada a objetos, llamada ZODB o Zope Object Database. Esta base de datos almacena objetos ordenados en un sistema similar a un sistema de ficheros, pero cada objeto tiene propiedades, métodos u otros objetos. Esta aproximación es muy

diferente de las base de datos relacionales habituales. Sin embargo, Zope dispone de múltiples conectores para las diferentes bases de datos relacionales y ofrece sistemas básicos de conexión y consulta abstrayéndolos como objetos. (<http://es.wikipedia.org/wiki/Zope>)

Apache

Apache, sustancialmente, es un proyecto nacido para crear un servidor de Web estable, fiable y veloz para plataformas Unix. Apache nace, por una parte, de un código ya existente y de una serie de parches (patch) para mejorar su fiabilidad y sus características; de ahí su nombre. (<http://www.ecured.cu/>)

Características:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este y están ahí para que se instalen cuando se necesiten. Otra cosa importante es que cualquiera que posea alguna experiencia en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
- Trabaja con Perl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y Apache utiliza su parte del pastel de Perl tanto con soporte CGI como con soporte mod perl. También trabaja con Java y páginas JSP. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto.

- Tiene una alta configurabilidad en la creación y gestión de logs. Permite la creación de ficheros de log a la medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

Justificación del Servidor Web a utilizar.

Para la creación del sistema se utilizó servidor Web Apache el cual es el complemento perfecto para páginas dinámicas desarrolladas con PHP y MySQL. Comparte con estos muchas de sus características, como son la gratuidad (también se trata de software libre), su popularidad, su sencillez de manejo y su versatilidad, ya que podemos instalarlo sobre Unix o sobre Windows. Es por ello que es utilizado en este proyecto. Su sencillez de manejo lo hace ideal para instalarlo en los ordenadores para hacer todo tipo de pruebas y ejercicios.

1.5.4 Paquete de tecnología Web

XAMPP - es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl. El nombre proviene de **X** (para cualquier sistema operativo), **A** (Apache), **M** (MySQL), **P** (PHP), **P** (Perl). El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X. (Blanco, 2008)

XAMPP es regularmente actualizado para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. Incluye otros módulos como OpenSSL, y phpMyAdmin.

Independientemente de que existen otras herramientas como son, los lenguajes de programación Web, gestores de bases de datos y servidores

Web, que, aunque no se mencionan y describen, no pierden su vigencia, funcionalidad e importancia, en esta investigación sólo se hace referencia:

- A los lenguajes de programación, Javascript y Personal Home Page (PHP).
- El gestor de bases de datos, MySQL.
- El paquete de tecnología Web, XAMPP.

1.5.5 Dreamweaver como un entorno de Desarrollo.

Adobe Dreamweaver es una aplicación en forma de estudio (basada en el Adobe Flash) enfocada a la construcción y edición de sitios y Aplicaciones Web basados en estándares.

Fue creado inicialmente por Macromedia (actualmente producido por Adobe Systems). Es el programa de este tipo más utilizado en el sector del diseño y la programación Web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Su principal competidor es Microsoft Expression Web y tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras. Hasta la versión MX, fue duramente criticado por su escaso soporte de los estándares de la Web, ya que el código que generaba era con frecuencia sólo válido para Internet Explorer, y no validaba como HTML estándar. Esto se ha ido corrigiendo en las versiones recientes. (<http://www.ecured.cu/>)

Facilidades que brinda

Dreamweaver permite al usuario utilizar la mayoría de los navegadores Web instalados en su ordenador para pre visualizar las páginas Web. También dispone de herramientas de administración de sitios dirigidas a principiantes como, por ejemplo, la habilidad de encontrar y reemplazar líneas de texto y

código por cualquier tipo de parámetro especificado, hasta el sitio Web completo. El panel de comportamientos también permite crear JavaScript básico sin conocimientos de código.

Con la llegada de la versión MX, Macromedia incorporó herramientas de creación de contenido dinámico en Dreamweaver. En lo fundamental de las herramientas HTML WYSIWYG, también permite la conexión a Bases de Datos como MySQL y Microsoft Access, para filtrar y mostrar el contenido utilizando tecnología de script como, por ejemplo, ASP (Active Server Pages), ASP.NET, ColdFusion, JSP (JavaServer Pages) y PHP sin necesidad de tener experiencia previa en programación. (<http://www.ecured.cu/>)

¿Por qué Dreamweaver?

La gran ventaja de este editor sobre otros es su gran poder de ampliación y personalización del mismo, puesto que en este programa, sus rutinas (como la de insertar un hipervínculo, una imagen o añadir un comportamiento) están hechas en Javascript-C, lo que le ofrece una gran flexibilidad en estas materias. Esto hace que los archivos del programa no sean instrucciones de C++ sino, rutinas de Javascript que hace que sea un programa muy fluido, que todo ello hace, que programadores y editores Web hagan extensiones para su programa y lo ponga a su gusto.

Las versiones originales de la aplicación se utilizaban como simples editores WYSIWYG. Sin embargo, versiones más recientes soportan otras tecnologías Web como CSS, JavaScript y algunos Framework del lado servidor.

Dreamweaver ha tenido un gran éxito desde finales de los 90 y actualmente mantiene el 90% del mercado de editores HTML. Esta aplicación está disponible tanto para la plataforma MAC como para Windows, aunque también se puede ejecutar en plataformas basadas en UNIX utilizando programas que implementan las API's de Windows, tipo Wine.

Como editor WYSIWYG que es, Dreamweaver oculta el código HTML de cara al usuario, haciendo posible que alguien no entendido pueda crear páginas y sitios Web fácilmente. (<http://www.ecured.cu/>)

1.5.6 EMS SQL Manager para MySQL.

Es una herramienta de alto rendimiento para la administración de MySQLDatabase Server y el desarrollo. Funciona con cualquier versión de MySQL desde la 3.23 a la más reciente y soporta todas las últimas características de MySQL, incluyendo disparadores, vistas, procedimientos almacenados y funciones, claves foráneas InnoDB, los datos Unicode y así sucesivamente. SQL Manager para MySQL permite crear / editar todos los objetos de base de datos MySQL, el diseño de bases de datos MySQL visualmente, ejecutar scripts de SQL, importar y exportar datos de bases de datos MySQL, gestión de usuarios de MySQL y sus privilegios, y tiene muchas otras características útiles para la eficiente administración de MySQL. SQL Manager para MySQL tiene una interfaz de estado de la técnica gráfica de usuario con el sistema de asistente bien descrito, de modo claro en su uso que ni un principiante no debe confundirse con ella. (<http://sqlmanager.net>)

¿Por qué EMS SQL Manager para MySQL?

Proponemos para la realización del software la utilización del Manager para MySQL porque este brinda una rápida y sencilla administración y navegación de las bases de datos además de gestión de seguridad y fácil creación de visuales.

Existen otras herramientas que no pierden su importancia y su vigencia en desarrollo de software, pero no son mencionadas en este documento debido a que el software se integrará a la intranet de la EMPLNI y se hace uso de las herramientas antes mencionadas, por política de la empresa.

1.6 Metodologías Existentes

En la actualidad el desarrollo del software está vinculado a las actividades realizadas para llevarlo a cabo, en un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Una metodología es un proceso.

No existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable. Es de vital importancia saber cuál es la metodología a utilizar teniendo en cuenta las características y las propiedades del producto que vamos a desarrollar. Las metodologías existentes en la actualidad se dividen en dos grandes grupos atendiendo a sus características: las metodologías tradicionales (RUP, MSF) y las metodologías ágiles (XP, SCRUM). Las primeras están pensadas para el uso exhaustivo de documentación durante todo el ciclo del proyecto mientras que las segundas ponen vital importancia en la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y al mantener una buena relación con el cliente. (Zulueta ,2010)

1.6.1 Metodologías Tradicionales

Persiguen la consecución de un proyecto cuya planificación está bien definida. Contando con los medios adecuados para controlar su ejecución, corregir posibles desviaciones y documentar lo realizado.

Proceso Unificado de Desarrollo (RUP).

Es un proceso para el desarrollo de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Como tres características esenciales está dirigido por casos de uso: que orientan al proyecto a la importancia para el usuario y lo que se quiere, está centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es iterativo e incremental: donde divide el proyecto

en mini-proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera depurada. RUP propone cuatro etapas para el desarrollo de un producto:

Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. Estas etapas revelan que para producir una versión del producto en desarrollo se emplean todas las actividades de ingeniería pero con diferente énfasis; en las primeras versiones se hace más énfasis en el modelado del negocio, requisitos, análisis y diseño; mientras en las posteriores el énfasis recae sobre las actividades de implementación, pruebas y despliegue. Además contempla flujos de trabajo de soporte que involucran actividades de planificación de recursos humanos tecnológicos y financieros.

Disciplina de Desarrollo:

- **Modelado empresarial:** Entendiendo las necesidades del negocio.
- **Requerimientos:** Traslado de las necesidades del negocio a un sistema automatizado.
- **Análisis y Diseño:** Traslado de los requerimientos dentro de la arquitectura de software.
- **Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
- **Despliegue:** Describe las actividades asociadas al garantizar que el producto de software esté disponible para los usuarios.
- **Gestión de cambios y configuración:** Explica cómo controlar y sincronizar la evolución del conjunto de productos de trabajo que componen un sistema de software.
- **Gestión de proyecto:** Planificación del proyecto, la gestión del riesgo, la supervisión del progreso y la métrica.

- **Entorno:** Organiza esos elementos de método que proporcionan el entorno de desarrollo de software que da soporte al equipo de desarrollo, incluidos los procesos y las herramientas.

Los elementos del RUP son:

Actividades: Son los procesos que se llegan a determinar en cada iteración.

Trabajadores: Son las personas o entes involucrados en cada proceso.

Artefactos: Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

(http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational)

1.6.2 Metodologías Ágiles

La comunidad de programadores se planteó utilizar un nuevo paradigma, radicalmente diferente al de las metodologías tradicionales. Metodologías más ágiles, capaces de lidiar con ese “entorno caótico” que es el desarrollo de software. Para así pasar a tener un “caos controlado” y poder llegar a resultados prácticos más fácilmente.

XP (Extreme Programming).

Es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las

metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

La programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica. Es una metodología ágil 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. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (<http://www.ecured.cu/>)

Procesos de XP

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.

3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del proyecto. (BECK, 1999)

Prácticas XP

La mayoría de estas prácticas no son nuevas, han sido reconocidas por la industria como mejores prácticas durante años. En la XP, dichas prácticas son llevadas al extremo para obtener más que la suma de las partes.

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas. (Beck, 1999)

El juego de la planificación. Hay una comunicación frecuente, el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

Entregas pequeñas. Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.

Metáfora.

El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).

Diseño simple. Se debe diseñar la más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. **Pruebas.** La producción de código está por las pruebas unitarias. Éstas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema

Refactorización (Refactoring). Es el proceso de modificar el código de un sistema de software modo que no se altere su comportamiento externo pero se mejore su estructura interna. Es una técnica disciplinada de reestructuración de código. Parte del proceso puede automatizarse y de hecho existen herramientas que facilitan la tarea.

Programación en parejas. Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores).

Propiedad colectiva del código. Cualquier programador cambiar cualquier parte del código en cualquier momento.

Integración continua. Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.

Cliente in-situ. El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.

Estándares de programación. XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible. (BECK, 1999). Las seis prácticas esenciales por donde se debe empezar:

Pruebas.

Refactorización.

Programación en pares.

Juego de planificación.

Liberaciones pequeñas.

Integración continua.

Fases de la metodología XP

Fase I: Planificación

- Se escriben historias de usuario, cuya idea principal es describir un caso de uso en dos o tres líneas con terminología del cliente (de hecho, se supone que deben ser escritos por el mismo), de tal manera que se

creen test de aceptación para historias de usuarios (user storie) y permita hacer una estimación de tiempo de desarrollo del mismo.

- Se crea un plan de lanzamiento (release planning), que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto.
- Se usa como base las historias de usuario, participando el cliente en la elección de las que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.
- El desarrollo se divide en iteraciones, cada una de las cuales comienzan con un plan de iteración, para el que se eligen las historias de usuario a desarrollar y las tareas de desarrollo.
- Se cambia el proceso cuanto sea necesario, para adaptarlo al proyecto.

Fase II: Diseño

- Se eligen los diseños funcionales más simples.
- Se elige una metáfora del sistema para que el nombrado de clases, siga una misma línea, facilitando la reutilización y la comprensión del código.
- Se escriben tarjetas de clase-responsabilidades-colaboración (CRC) para cada objeto, que permitan abstraerse al pensamiento estructurado y que el equipo de desarrollo completo participe en el diseño.

Fase III: Codificación

- El cliente está siempre disponible, de ser posible, cara a cara. La idea es que forme parte del equipo de desarrollo, y esté presente en todas las fases de XP. La idea es usar el tiempo del cliente para estas tareas en lugar de crear una detallada especificación de requisitos, y evitar la entrega de un producto insuficiente, que le hará perder tiempo.

- El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código.
- Las pruebas unitarias se codifican antes que el código en sí, haciendo que la codificación de este último sea más rápida, y que cuando se afronte la misma se tenga más claro, qué objetivos tiene que cumplir lo que se va a codificar.
- La programación del código se realiza en parejas, para aumentar la calidad del mismo. En cada momento, sólo habrá una pareja de programadores que integre código.
- Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo y permitiendo que todo el mundo trabaje con la última versión del desarrollo. De esta manera, se evitará pasar grandes períodos de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades serán detectadas enseguida.
- Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentar la contribución de ideas por parte de todo el equipo de desarrollo.
- Se deja la optimización para el final.
- No se hacen horas extra de trabajo.

Fase IV: Pruebas

- Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado.
- Cuando se encuentra un error de codificación o bug, se desarrollan pruebas para evitar volver a caer en el mismo.
- Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las historias de usuarios elegidas para la iteración, y son "pruebas de caja negra", en las que el cliente verifica el correcto funcionamiento de lo

que se está probando. Cuando se pasa la prueba de aceptación, se considera que la correspondiente historia de usuario se ha completado.

Justificación de la metodología a utilizar.

¿Por qué XP?

Actualmente, XP es el método ágil más documentado (hay una colección de libros con “XP Series” de Addison Wesley) y extendido. Existe una gran comunidad de desarrolladores XP. Otra de las ventajas de XP es que no es necesario adoptarlo en forma completa, sino que pueden utilizarse varias de sus prácticas en forma independiente. Esto hace que el costo de su implementación sea mucho más accesible que el de otras metodologías.

Un estudio a la bibliografía, muestra las ventajas y desventajas que tiene XP y que exponemos en los dos puntos siguientes.

Ventajas de XP

- Puede ser implementado en forma parcial (elegir sólo algunas de las prácticas)
- Puede ser implementado en forma gradual
- Puede adaptarse a las necesidades de cualquier equipo de desarrollo. De hecho, Kent Beck recomienda a los equipos que lo adapten a sus necesidades.
- Exige que se establezca una comunicación más fluida con el cliente y que este tenga mayor participación en el proceso de desarrollo. La consecuencia de esto es que el cliente se involucre más en el desarrollo del producto.
- Actualmente es la metodología ágil más extendida y documentada
- Se realizan pruebas constantemente del sistema (BECK, 1999).

1.7 Estilo Arquitectónico:

Define las reglas generales de la organización en términos de un patrón y las restricciones en la forma y la estructura de un grupo numeroso.

Arquitectura en capas:

Es donde se define como organizar el modelo de diseño a través de capas, que pueden estar físicamente distribuidas, lo que quiere decir que los componentes de una capa solo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización de sistemas complejos, reduciendo dependencias de forma que las capas más bajas no son consistentes de ningún detalle o interfaz de las superiores.

La programación por capas es un estilo de programación en el que el objetivo primordial es separar la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación del usuario.

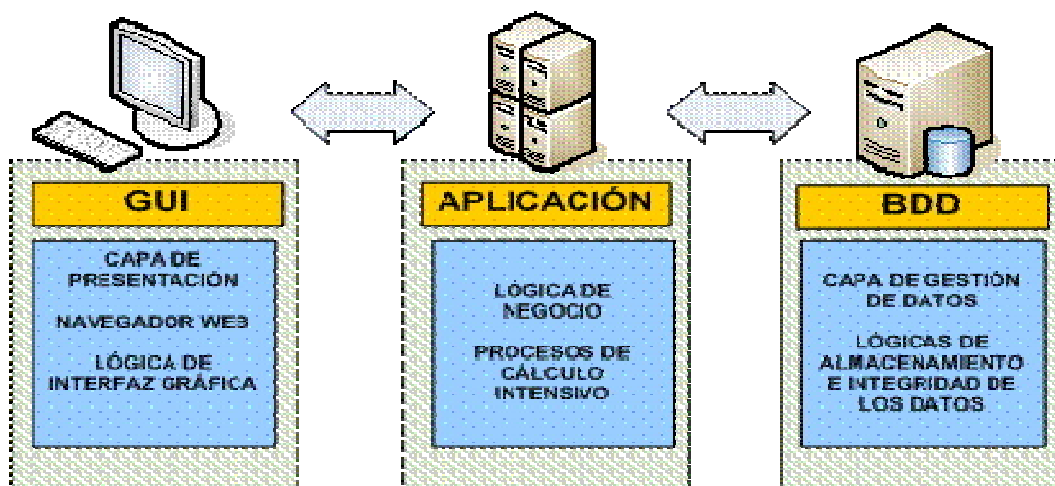


Fig.1.1 Arquitectura en capas

Capas o niveles.

- **Capa de presentación o interface:** es la capa que le permite al usuario interactuar con el sistema, captura y le comunica la información al mismo, dando un mínimo de proceso (realiza un filtrado previo para asegurarse que no haya errores de formato). Esta capa se comunica únicamente con la de negocio.
- **Capa de negocio:** es donde reside los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitarle al gestor de bases de datos almacenar o recuperar datos de él.
- **Capa de datos:** es donde residen los datos y es la encargada de acceder ellos. Está formada por uno o más gestores de bases de datos que se encargan de realizar el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un mismo ordenador aunque no es lo típico. Lo más usual es que haya una multitud de ordenadores donde reside la capa de interface (son los clientes de la arquitectura cliente/servidor). Las capas de negocios y de datos pueden residir en un mismo ordenador, y si el crecimiento de las necesidades lo aconseja, pueden dividirse en dos o más ordenadores. Así, si el tamaño o complejidad de las base de datos aumenta, pueden dividirse en varios ordenadores los cuales recibirán las peticiones del ordenador donde resida la capa de negocio. Si por el contrario, la complejidad fuese en la capa de negocio lo que obligase a la separación, esta lógica del negocio podría residir en uno o más ordenadores que realizarían las solicitudes a una única base de datos.

(<http://kernelerror.net/programacion/php/arquitectura-3-capas>)

Modelo Vista Controlador (MVC)

Para el diseño de aplicaciones con sofisticadas interfaces se emplea el patrón de diseño MCV. La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si se realiza un diseño ofuscado, es decir, una forma de mezclar los componentes de interfaz y de negocio, entonces, la consecuencia será que, cuando se necesite cambiar la interfaz, tendrá que modificarse trabajosamente los componentes de negocio, por lo que propiciará mayor trabajo y más riesgo de error.

Se trata de realizar un diseño que desacople la vista del modelo, con el fin de perfeccionar la reusabilidad. De este modo las modificaciones son las que impactan en menor medida en la lógica de negocio o de datos.

Se originó en la comunidad Smalltalk para implementar interfaces de usuario en los que las responsabilidades están bien distribuidas entre distintas partes (componentes) del diseño.

Así, se decidió, distinguir tres responsabilidades distintas:

- **Modelo:** Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- **Vista:** Presenta el modelo en un formato adecuado, como en una página Web que le permite al usuario interactuar con ella, usualmente un elemento de interfaz de usuario.
- **Controlador:** Responde a eventos, usualmente acciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Justificación de la arquitectura a utilizar

Proponemos como estilo arquitectónico a utilizar en nuestro proyecto la arquitectura en tres capas debido a las ventajas que esta brinda:

Ventajas de esta Arquitectura

- El desarrollo se puede llevar a cabo en varios niveles.
- Desarrollos paralelos (en cada capa).
- Aplicaciones más robustas debido al encapsulamiento.
- En caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad)
- Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buen escalado, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad.
- Las llamadas de la interfaz del usuario en la estación de trabajo, al servidor de capa intermedia, son más flexibles que en el diseño de dos capas, ya que la estación solo necesita transferir parámetros a la capa intermedia.
- Con la arquitectura de tres capas, 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 formato modular. La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.

1.8 Conclusiones

En este capítulo se trataron temas esenciales para el desarrollo de la aplicación sin los cuales hubiera sido imposible la materialización del objeto de estudio y el campo de acción. Además se realizó un estudio para la selección de las herramientas, metodología y lenguajes a utilizar dada la gama de ventajas que presentan.

CAPITULO II - PLANIFICACIÓN Y DISEÑO

2.1 Introducción

En este capítulo, se pone en práctica la fase de planeación y diseño, en la cual se determinan y detallan a fondo las necesidades del cliente, y se plantean los requisitos funcionales y no funcionales del sistema propuesto, los mismos serán implementados mediante el empleo de las historias de usuarios (HU), para lo cual será necesaria una estimación del esfuerzo necesario en la realización de las mismas y el establecimiento del plan de iteraciones del sistema y su duración, además del plan de entrega del proyecto, se definirá también la creación de las tarjetas CRC (Clases-Responsabilidades-Colaboración) la cuales son un paradigma de la programación orientada a objetos y se hará una descripción de las interfaces de usuario.

2.2. Personal relacionado con el sistema

Personas relacionadas con el sistema	Justificación
Esp.B en Gestión de RR HH	Persona encargada de elaborar los perfiles y sus competencias.
Comisión Evaluadora	Personas encargadas de determinar las necesidades de capacitación de cada trabajador a partir de su perfil de competencias.
Capacitadora	Es la encargada de elaborar el plan de capacitación de la entidad con las acciones correspondientes para cada trabajador por área.
Administrador	Es la persona encargada de asesorar y dar seguimiento al estado del

	proceso de desarrollo.
Desarrollador	Es la persona responsable de llevar a cabo la implementación del sistema.

Tabla 2.1 Personas relacionadas con el sistema

2.3 Funcionalidades

Una vez descritas las personas relacionadas con el sistema y los requisitos generales del sistema, se describe a continuación una lista con las funcionalidades del producto, donde se especifican las funcionalidades que el sistema debe cumplir y las cualidades que el producto debe tener. Dicha lista puede sufrir modificaciones a partir del conocimiento que se va obteniendo del producto, los datos que el cliente facilita y a partir de las necesidades de este. Dicha lista presenta como finalidad que al terminar el software este sea lo más correcto y útil.

2.3.1 Requisitos funcionales

Número	Descripción
1.	Insertar Perfil.
2.	Listar Perfil.
3.	Modificar Perfil.
4.	Eliminar Perfil.
5.	Ver generalidades del Perfil.
6.	Insertar Competencias.
7.	Listar Competencias.
8.	Eliminar Competencias.
9.	Modificar Competencias.
10.	Agregar Competencias a un Perfil.
11.	Eliminar Competencias de un Perfil.

12.	Asignar importancia a competencias de un perfil.
13.	Insertar trabajador.
14.	Listar trabajador.
15.	Eliminar trabajador
16.	Modificar trabajador.
17.	Asignar títulos a un trabajador.
18.	Ver títulos de un trabajador.
19.	Eliminar títulos de un trabajador.
20.	Modificar títulos de un trabajador.
21.	Insertar evaluación de competencias.
22.	Mostrar evaluación de competencias.
23.	Modificar evaluación de competencias.
24.	Determinar evaluación final y necesidad de capacitación.
25.	Mostrar evaluación final y necesidad de capacitación.
26.	Insertar acción.
27.	Mostrar acción.
28.	Modificar acción.
29.	Eliminar acción.
30.	Asignar acciones a un trabajador.
31.	Eliminar acciones de un trabajador.

Tabla 2.1 Requisitos Funcionales

2.3.2 Requisitos no funcionales

Descripción:

Apariencia Interfaz Externa:

- Diseño orientado a llamar la atención del usuario y con una navegación sencilla.

-
- Construcción de enlaces rápidos o anclas para los documentos muy largos.

Usabilidad:

- Facilidad de uso por parte de los usuarios: el sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.
- Especificación de la terminología utilizada: el sistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vista a una mayor comprensión por parte del cliente de la herramienta de trabajo.
- Menús: el sistema debe presentar una serie de menús tanto laterales como en barra de iconos flotantes que permitan el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de estas estructuras.

Rendimiento:

- Los tiempos de respuesta no son determinantes.

Portabilidad:

- Necesidad de que el sistema sea multiplataforma.

Seguridad:

- Identificar al usuario antes de que pueda realizar cualquier operación sobre el contenido del sistema.
- Garantizar que la información sea editada solo por quien tiene derecho a editarla.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de acceso del usuario activo.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Verificación sobre acciones irreversibles (eliminaciones).

Software:

- Navegador compatible o superior con Internet Explorer4 y Mozilla Firefox 4.0 Beta 7
- Apache 2.0
- PHP 5.0
- MySQL 5.2

Requisitos para la documentación de usuarios en línea y ayuda del sistema:

- Manual de usuario: el sistema deberá presentar un manual de usuario, permitiendo con ello un correcto uso de sus funcionalidades y brindarle al usuario una mayor experiencia del trabajo con el mismo.
- Documentación actualizada del grupo de desarrollo: se precisa que la documentación del sistema esté actualizada en todos los aspectos, fases de trabajo y ciclos de desarrollo del mismo, permitiendo con ello un respaldo legal del desarrollo de dicho sistema.

Interfaces de Comunicación:

- El sistema presta enlaces de comunicación con otros entornos web de la red como
-

-
- (Granma, Ecured, Trabajadores y Juventud Rebelde).
 - La comunicación entre el cliente y el servidor de aplicaciones es a través del protocolo HTTP.
-

Tabla 2.2 Requisitos no Funcionales

2.4 Historias de usuario

Las HU, son la técnica utilizada en XP para detallar los requisitos del software. Son el resultado directo del intercambio entre los usuarios y desarrolladores a través de reuniones donde las conocidas *tormenta de ideas (brain storm)* arrojan no solo los requerimientos, sino también las posibles soluciones; representan una forma rápida de administrar las necesidades de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para gestionarlos, debido a que un requerimiento de software es descrito de forma concreta y sencilla utilizando el lenguaje común del usuario. Las HU permiten responder ágilmente a los requerimientos cambiantes y aunque se redactan desde las perspectivas de los clientes, también los desarrolladores pueden brindar ayuda en la identificación de las mismas. Para definir las se emplea la siguiente plantilla.

Modelo de planilla de historia de usuario

Historia de usuario	
Número: No. Historia de usuario	Usuario: Usuario entrevistado para obtener información.
Nombre: nombre para identificar la historia de usuario.	
Prioridad en el negocio: Importancia: Alta / Media / Baja	Riesgo en desarrollo: Dificultad: Alta / Media / Baja
Puntos estimados: Estimación: de 1 a 3 puntos	Iteración asignada: Iteración a la que corresponde
Programador responsable: Nombre de encargado de programación.	

Descripción: Breve descripción de lo que realizará la HU.
Observaciones: Algunas observaciones de interés.

Tabla 2.3 Planilla de historia de usuario

HU No.1: Gestión de perfiles

Historia de Usuario	
Número: 1	Usuario: Esp.B en Gestión de RR HH
Nombre de historia: Gestión de Perfiles.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 3	Iteración Asignada: 1
Programador responsable: Leonart Urgellés Olivero.	
Descripción: El usuario debe insertar los datos de los Perfiles. Una vez añadidos, se podrá mostrar, actualizar, eliminar y buscar la información referente a ellos.	
Observaciones: Confirmado con el cliente.	

Tabla 2.4 HU No.1: Gestión de perfiles

Para ver las historias de usuario ver [\[ANEXO 1: HISTORIAS DE USUARIO\]](#).

2.5 Planificación de entregas

En esta fase se establece la prioridad de cada HU, y a continuación, se realiza una estimación del esfuerzo necesario de cada una de ellas por parte de los programadores. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debe obtenerse en no más de dos a tres meses.

Las estimaciones asociadas a la implementación de las historias se establecen empleando como medida el punto de estimación. Un punto de estimación equivale a una semana ideal de programación, donde los miembros de los equipos de desarrollo, trabajan el tiempo planeado sin ningún tipo de

interrupción, este punto de estimación que se utiliza para representar la semana ideal, es de 5 días. Las historias generalmente tienen un valor de 1 a 3 puntos. Además, se mantiene un registro de la velocidad de desarrollo, establecida por puntos de iteración, basado fundamentalmente en la suma de los puntos de estimación correspondientes a las HU, que fueron terminadas en la última iteración.

2.5.1 Estimación de esfuerzo por historias de usuario

Para el buen desarrollo del sistema propuesto, se realizó una estimación para cada una de las HU identificadas y se obtienen los resultados que se muestran a continuación:

Estimación de esfuerzo por historia de usuario

Historias de usuario	Puntos de estimación
Gestión de perfiles	3 semanas
Gestión de competencias	2 semanas
Gestión de competencias del Perfil	2 semanas
Gestión de trabajadores	3 semanas
Gestión de títulos	2 semanas
Gestión de evaluaciones	3 semanas
Gestión de acciones	3 semanas

Tabla 2.11 Estimación de esfuerzo por historia de usuario

2.5.2 Planificación de iteraciones

A partir de las HU antes expuestas y la estimación del esfuerzo propuesto para la realización de las mismas, se procede a realizar la planificación de la etapa de implementación del sistema, apoyándose en el tiempo e intentando concentrar las funcionalidades relacionadas en una misma iteración. En este plan se establece cuántas iteraciones serán necesarias realizar sobre el

sistema para su terminación. El plan de iteraciones puede contener indicaciones sobre cuáles HU se incluirán en un release, lo cual debe ser consistente con el contenido de una o dos iteraciones.

En relación con lo antes tratado se decide realizar el sistema en 4 iteraciones, las cuales se explican de forma detalla a continuación:

Primera iteración:

Esta iteración tiene como objetivo darle cumplimiento a las HU que se consideraron de mayor importancia para el desarrollo de la aplicación ya que de estas depende el desarrollo de las posteriores. Al concluir dicha iteración se contará con todas las funcionalidades descritas en las HU 1,2 y 3 las cuales hacen alusión a la gestión de los perfiles de competencias y las competencias de estos. Además se tendrá la primera versión de prueba, que contará con dos modelos de desarrollo que incorporan todas las funcionalidades antes vistas, estos modelos se presentarán al cliente con el objetivo de obtener una retroalimentación del mismo para posteriores iteraciones del producto.

Segunda iteración:

Esta iteración tiene como finalidad desarrollar las HU 4 y 5. Las mismas son las que brindan las funcionalidades de la gestión de los trabajadores y sus títulos a partir de la iteración anterior. La versión que se obtenga de esta iteración en unión con la entregada en la iteración anterior se le facilitará al cliente una oportunidad para comprobar si cumple con las necesidades antes acordadas con él.

Tercera iteración:

Esta iteración del módulo tiene como propósito llevar a cabo el desarrollo de la HU 6 y 7. La cuales proporcionan las funcionalidades de gestionar las

evaluaciones y acciones de los trabajadores. Además de determinar si un trabajador necesita o no capacitación.

2.5.3 Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto guiado por la metodología de desarrollo de software XP, se crea el plan de duración de cada una de las iteraciones que se llevarán a cabo durante el desarrollo del mismo. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las HU en cada una de las mismas.

Plan de duración de las iteraciones

Iteración	Historias de usuario	Duración total
Iteración 1	Gestión de perfiles	7 semanas
	Gestión de competencias	
	Gestión de competencias del perfil.	
Iteración 2	Gestión de trabajadores	5 semanas
	Gestión de títulos	
Iteración 3	Gestión de evaluaciones	6 semanas
	Gestión de acciones	

Tabla 2.12 Plan de duración de las iteraciones

2.6 Tarjetas Clases-Responsabilidades-Colaboración (CRC).

El uso de las tarjetas C.R.C (Clases, Responsabilidades y Colaboración) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica. Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad. Esta nueva

técnica de diseño es adoptada como alternativa a los diagramas UML de las clases, pues en estas se plasman las responsabilidades que tienen cada objeto y las clases con las que tienen que interactuar para darles respuesta brindando así la información que se necesita a la hora de implementar(Beck 2004).

HU No.1: Gestión de perfiles

Nombre de la clase: Gestion _ perfiles	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Insertar datos de Perfiles.	Perfiles, Connections
Listar los Perfiles.	Perfiles, Connections
Modificar los Perfiles.	Perfiles, Connections
Eliminar los Perfiles.	Perfiles, Connections
Ver generalidades del Perfil.	Perfiles, Connections

Tabla 2.13 Gestión de Perfiles

Para ver las tarjetas CRC ver [\[ANEXO 2: TARJETAS CRC\]](#).

2.7 Descripción de las interfaces de usuario.

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar.

Las interfaces básicas de usuario son aquellas que incluyen elementos como menús, ventanas, teclado, ratón, los *beeps* y algunos otros sonidos que la computadora hace, y en general, todos aquellos canales por los cuales se permite la comunicación entre el ser humano y la computadora. La mejor

interacción humano-máquina a través de una adecuada interfaz (Interfaz de Usuario), que le brinde tanto comodidad, como eficiencia. (http://es.wikipedia.org/wiki/Interfaz_de_usuario)

Para el desarrollo del software fue diseñada una interfaz de usuario sencilla que combinara con los colores emblemáticos de la empresa y que proporcionara además las funcionalidades para las que fue creada la aplicación, teniendo en cuenta que la intranet de la entidad está desarrollada bajo el mismo prototipo.

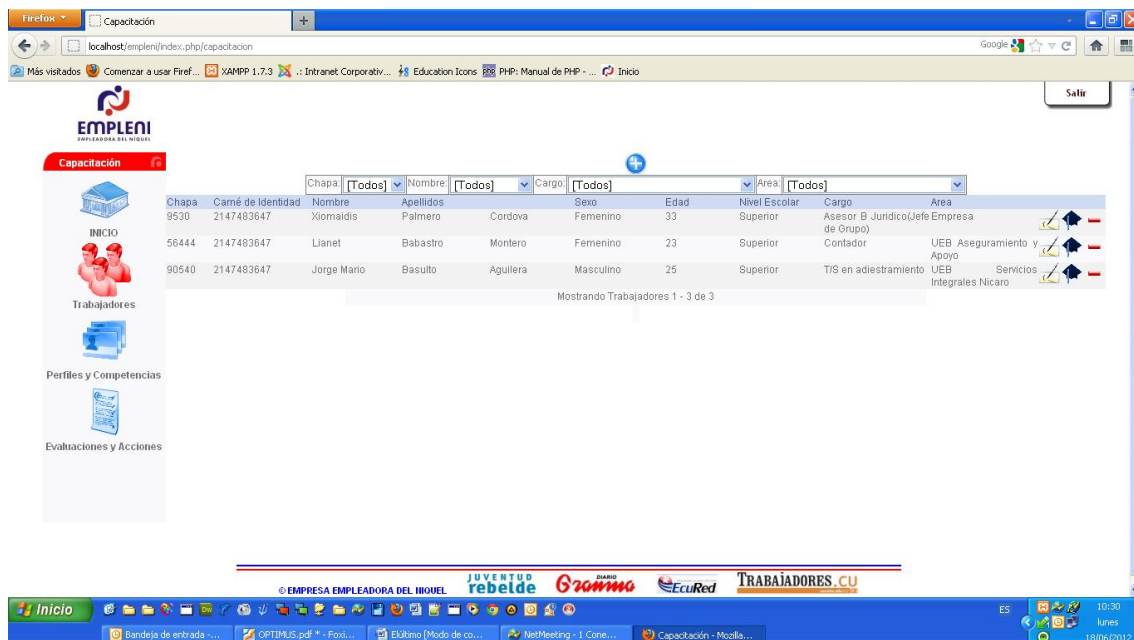


Fig.2.2 Lista de trabajadores

Para ver las Interfaces de Usuario ver [\[ANEXO 3: INTERFACES DE USUARIO\]](#).

2.8 Conclusiones.

En este capítulo se fundamentaron los aspectos referentes a la fase de planeación y diseño, teniendo en cuenta la participación y las peticiones del cliente, se elaboraron las HU y se estimó el esfuerzo necesario para la realización de cada una de ellas. Con lo cual se conformó el plan de lanzamiento de las iteraciones teniendo en cuenta que para pasar de una iteración a otra debe haberse cumplido la anterior. Dándose por terminada esta fase de la metodología y con la base para pasar a la siguiente etapa de desarrollo.

CAPITULO III – DESARROLLO Y PRUEBAS**3.1 Introducción**

Teniendo en cuenta que ya fue concluida la fase de Planificación y Diseño en este capítulo se le da inicio a la fase de Desarrollo y Pruebas conforme a la metodología utilizada. Por otra parte se describen cada una de las tareas confeccionadas para cumplir con el desarrollo de cada una de las HU definidas. Además se mostrarán las pruebas de aceptación confeccionadas por el cliente para comprobar que la aplicación funcione correctamente. Estas pruebas fueron realizadas durante la entrega que se efectuaban a lo largo del desarrollo del proyecto. Se expone el modelo de datos empleado para la aplicación y se desarrollan las iteraciones a partir de las HU.

3.2 Desarrollo de las iteraciones

Durante la fase planificación y diseño fueron detalladas las HU correspondientes a cada una de las iteraciones a desarrollar, teniendo en cuenta las prioridades y restricciones de tiempo, previstas por el cliente.

3.2.1 Tareas por historias de usuario

Dentro del contenido de este plan, las HU se descomponen en tareas de programación o ingeniería, y a su vez, estas son asignadas al equipo de desarrollo para su implementación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, sólo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Las mismas se representan mediante las tarjetas de tareas.

Distribución de tareas por historia de usuario

Historia de usuario	Tareas
Gestión de perfiles	• Insertar perfil • Listar perfil • Modificar perfil

	• Eliminar perfil • Ver generalidades del perfil.
Gestión de competencias	• Insertar competencias. • Listar competencias. • Eliminar competencias. • Modificar competencias.
Gestión de competencias del perfil	• Agregar competencias a un perfil. • Eliminar competencias de un perfil. • Asignar importancia a competencias de un perfil.
Gestión de trabajadores	• Insertar trabajador. • Listar trabajador. • Eliminar trabajador. • Modificar trabajador.
Gestión de títulos	• Asignar títulos a un trabajador. • Ver títulos de un trabajador. • Eliminar títulos de un trabajador. • Modificar títulos de un trabajador.
Gestión de Evaluaciones	• Insertar evaluación de competencias. • Mostrar evaluaciones de competencias. • Modificar evaluaciones de competencias. • Determinar evaluación final y necesidad de capacitación. • Mostrar evaluación final y necesidad de capacitación.
Gestión de Acciones	• Insertar acciones. • Mostrar acciones. • Modificar acciones. • Eliminar acciones. • Asignar acciones a un trabajador. • Eliminar acciones de un trabajador.

Tabla 3.1 Distribución de tareas por historia de usuario

Historias de usuario abordadas en la primera iteración

Historias de usuario	Tiempo de estimación (semanas)	
	Estimación inicial	Real
Gestión de perfiles	3	2.5
Gestión de competencias	2	2
Gestión de competencias del perfil	2	1.2

Tabla 3.2 Historias abordadas en la primera iteración

Historias de usuario abordadas en la segunda iteración

Historias de usuario	Tiempo de estimación (semanas)	
	Estimación inicial	Real
Gestión de trabajadores	3	2
Gestión de títulos	2	1.5

Tabla 3.14 Historias de usuario abordadas en la segunda iteración

Historias de usuario abordadas en la tercera iteración

Historias de usuario	Tiempo de estimación (semanas)	
	Estimación inicial	Real
Gestión de Evaluaciones	3	2.3
Gestión de Acciones	3	3

Tabla 3.23 Historias de usuario abordadas en la tercera iteración

Para ver las tarjetas de ingeniería ver [[ANEXO 4: TARJETAS DE TAREAS](#)].

3.3 Modelado de datos

A continuación se presentan las tablas con que interactúa el sistema propuesto en esta investigación; se recogen y modelan todos los datos que dispone la aplicación.

Como resultado del modelado de datos se obtuvieron nueve tablas, que relacionadas entre sí responden a las solicitudes de recuperación y almacenamiento de datos necesarias para el funcionamiento de la aplicación.

Las mismas cumplen con las restricciones de la segunda forma normal (2NF) y Ningún atributo no-primario de las tablas es dependiente transitivamente de una clave primaria por lo que queda demostrado que está normalizada en tercera forma normal.

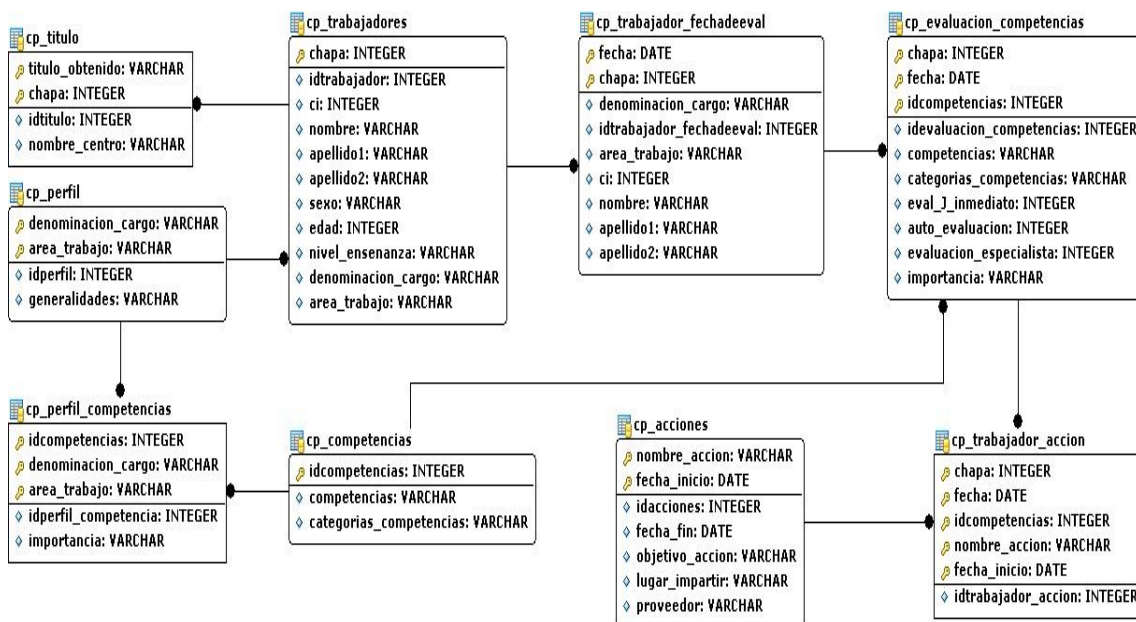


Fig.3.1 Modelo de datos

3.4 Pruebas

Una de las características de la metodología XP es el proceso de pruebas. Al realizar pruebas al software tanto como sea posible se aumenta 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.

3.4.1 Pruebas de aceptación

XP propone la realización de pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente.

Planilla de prueba de aceptación

Prueba de aceptación
HU: Nombre de la historia de usuario que va a comprobar su funcionamiento.
Nombre: Nombre del caso de prueba.
Descripción: Descripción del propósito de la prueba.
Condiciones de ejecución: Precondiciones para que la prueba se realice.
Entrada/Pasos ejecución: Pasos para probar la funcionalidad.
Resultado: Resultado que se desea de la prueba.
Evaluación de la prueba: Aceptada o denegada.

Tabla 3.34 Planilla de prueba de aceptación.

Pruebas de aceptación para la HU No.1: Gestión de perfiles

Prueba de aceptación
HU: 1
Nombre: Gestión de perfiles.
Descripción: La especialista debe suministrarle al sistema los datos de los perfiles para luego mostrarlos, modificarlos, eliminarlos y ver sus generalidades.
Condiciones de ejecución: El usuario debe introducir los datos de los perfiles.
Entrada/Pasos ejecución: • Insertar el perfil • Ver el listado de perfiles • Modificar un perfil • Eliminar un perfil • Ver generalidades de un perfil
Resultado Esperado: Gestionar correctamente el perfil
Evaluación de la prueba: Aceptada.

Tabla 3.35 PA: Prueba para comprobar la gestión de perfiles.

Para ver las pruebas de aceptación ver [\[ANEXO 5: PRUEBAS DE ACEPTACION\]](#).

3.5 Conclusiones.

En este capítulo se llevó a cabo la fase de desarrollo y pruebas, se realizó el desarrollo de las iteraciones a partir de la distribución de tareas por HU, y se les practicó las pruebas de aceptación a las funcionalidades. Se presentó el modelo de datos de la aplicación obtenida, logrando una visión detallada de sus atributos y las relaciones entre sus clases.

CAPITULO IV – ESTUDIO DE FACTIBILIDAD

4.1 Introducción

Después de definir la problemática existente y establecer las causas que ameritan de un nuevo sistema es pertinente realizar un estudio de factibilidad para determinar la infraestructura tecnológica y la capacidad técnica que implica la implantación del sistema en cuestión, así como los costos, beneficios y el grado de aceptación que la propuesta genera en la institución. Este análisis permitió determinar las posibilidades de diseñar el sistema propuesto y su puesta en marcha, los aspectos fueron clasificados en tres áreas, las cuales se describen a continuación:

- Factibilidad técnica.
- Factibilidad económica.
- Factibilidad operativa.

4.2 Estudio de factibilidad

4.2.1 Factibilidad técnica

La Factibilidad Técnica consistió en realizar una evaluación de la tecnología existente en la organización, este estudio estuvo destinado a recolectar información sobre los componentes técnicos que posee la organización y la posibilidad de hacer uso de los mismos en el desarrollo e implementación del sistema propuesto y de ser necesario, los requerimientos tecnológicos que deben ser adquiridos para el desarrollo y puesta en marcha del sistema en cuestión.

De acuerdo a la tecnología necesaria para la implantación del Sistema Informático para gestión de información de capacitación en la EMPLNI, se evaluó bajo dos enfoques: Hardware y Software.

Hardware

En la siguiente tabla se muestran la descripción de los requerimientos mínimos óptimos que debe cumplir el servidor donde debe estar el sistema y los disponibles en la organización:

Requerimientos Mínimos Óptimos	Disponibles
• Procesador 1.6Mhz	• Dual Core 2.6Mhz
• 512 Mb de Memoria RAM	• 2 GB
• Disco Duro de 20 GB	• 465.8 GB
• Tarjeta de Red	• Tarjeta de Red
• Tarjeta de Video	• Tarjeta de Video
• Monitor, Teclado, Mouse y Unidad de Protección UPS	• Monitor, Teclado, Mouse y Unidad de Protección UPS

Tabla 3.42 Descripción de requerimientos.

Evaluando el Hardware existente y tomando en cuenta la configuración mínima necesaria, la institución no requirió realizar inversión inicial para la adquisición de nuevos equipos, ni tampoco para repotenciar o actualizar los equipos existentes, ya que los mismos satisfacen los requerimientos establecidos tanto para el desarrollo y puesta en funcionamiento del sistema propuesto.

Software

En cuanto al Software la institución cuenta con todas las aplicaciones que se emplearon para el desarrollo del proyecto y funcionamiento del sistema, lo cual no amerita inversión alguna para la adquisición de los mismos. Las estaciones de trabajo operan bajo ambiente Windows. Para el uso general de las estaciones en actividades diversas se debe poseer las herramientas de escritorio y los navegadores que existen en el mercado actualmente.

Como resultado del estudio técnico se determinó que en los momentos actuales, la institución posee la infraestructura tecnológica (Hardware y Software) necesaria para el desarrollo y puesta en funcionamiento del sistema propuesto.

4.2.2 Factibilidad económica

A continuación se presenta un estudio que dio como resultado la factibilidad económica del desarrollo del nuevo sistema de capacitación. Se determinaron los recursos para desarrollar, implantar y mantener en operación el sistema programado, haciendo una evaluación donde se puso de manifiesto el equilibrio existente entre los costos del sistema y los beneficios que se liberaron de este, lo cual permitió observar de una manera más precisa las bondades del sistema propuesto.

4.2.2.1 Efectos económicos

Efectos directos:

Positivos

1. El personal relacionado con el sistema:

- El administrador ahorraría tiempo y recursos en cuanto a la instalación y mantenimiento del software ya que la instalación del mismo se efectuaría sobre la intranet del centro en su mismo formato, estilo y tecnologías utilizadas.
- Los demás usuarios del sistema (especialistas, comisiones y capacitadora del Dpto. de R/H de la empresa) tendrán la posibilidad de realizar de forma integrada sus actividades de capacitación y de acceso a la información necesaria para cumplir sus actividades.

3. Se establece un mecanismo eficaz y eficiente de control del proceso de capacitación en la empresa.

4. Se obtiene información actualizada y antigua, resumida y detallada, del proceso desarrollado.
5. Se integra la gestión del conocimiento de capacitación en el centro a la intranet de la entidad.

Para tener una idea de estos efectos se decidió distribuir su evaluación en dos momentos detallados en la tabla siguiente:

Situación con el producto anterior	Situación con el producto actual
El administrador instala la aplicación en cada una de las máquinas de los usuarios.	El administrador instala la aplicación en el servidor de la intranet, el usuario solo accede a ella y dentro de este al módulo capacitación.
El proceso de mantenimiento y actualización del software es más complicado debido a que esta se debe distribuir usuario por usuario.	El mantenimiento del software es más sencillo debido a que la aplicación está desarrollada con las mismas características de la intranet.
El trabajo de actualización aunque puede ser llevado a cabo por varios programadores después hay que agrupar los cambios en un producto final.	El trabajo de actualización del software puede ser realizado por varias personas a la vez sin tener que agrupar los cambios luego.
La elaboración de un plan de trabajo demoraba más tiempo.	Reducción del tiempo de elaboración del plan de trabajo.

Tabla 3.43 Situaciones con el producto anterior y con el producto actual

Negativos

1. Navegador compatible o superior con Internet Explorer 4 y Mozilla Firefox 4.0 Beta 7, solo bajo estos navegadores se podrá ver los estilos de diseños de la aplicación, debido a que las versiones inferiores no cuentan con los estándares de diseño del proyecto.

Efectos indirectos:

Debido a que este software no está construido con finalidad de venta sus efectos económicos indirectos no son de un valor significativo en el mercado.

Efectos externos:

El software desarrollado tendrá efectos sobre otras áreas de la entidad ya que la información con que se trabaja agilizará o afectará en dependencia de la velocidad, algunos procesos o actividades de la entidad.

4.2.2.2 Fichas de costo

Para determinar el costo económico del proyecto se elaboraron las fichas de costo correspondientes a las monedas libremente convertible y nacional.

Costos en moneda libremente convertible

Ficha de Costo.		
		Precio(s)
Costos Moneda Libremente Convertible		
Costos Directos		
Compra de equipos de cómputo		0,00
Alquiler de equipos de cómputo		0,00
Compra de licencia de Software		0,00
Materiales directos		0,00
Subtotal		0,00
Costos Indirectos		
Formación del personal que elabora el proyecto		0,00
Gastos en llamadas telefónicas		0,00

CAPITULO IV – ESTUDIO DE FACTIBILIDAD

Gastos para el mantenimiento del centro	0,00
Know How	0,00
Gasto por consumo de energía eléctrica	19,58
Gastos en representación	0,00
Subtotal	19,58
Gastos de Distribución y Venta	
Participación en ferias o exposiciones	0,00
Gastos en transportación	0,00
Compra de materiales de propagandas	0,00
Subtotal	0,00
Total	19,58

Tabla 3.44 Ficha de costo en Moneda Librementemente Convertible.

Costos en Moneda Nacional

Ficha de Costo.		
		Precio(s)
Costos Moneda Nacional		
Costos Directos		
Salario del personal que laborará en el proyecto		400,00
Depreciación de equipos		462,77
Seguridad Social		0,00
Vacaciones $100 \times 9,09\% = 9.09$		0,00
Gastos en llamadas telefónicas		0,00
Impuesto por la Fuerza de Trabajo		0,00
Subtotal		862,77
Costos Indirectos		
Know How		0,00
Subtotal		
Total		862,77

Tabla 3.45 Ficha de costo en Moneda Nacional.

El análisis de costo-beneficio se basa en un principio muy simple: compara los beneficios y los costos de un proyecto particular y, si los primeros exceden a los segundos entrega un elemento de juicio inicial que indica su aceptabilidad, mientras que el análisis costo-efectividad aunque sigue la misma lógica, compara los costos con las potencialidades de alcanzar más eficientemente los objetivos no expresables en moneda, sino en productos. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo el tiempo en minutos empleado para realizar las actividades de capacitación en la empresa y la variable sería la complejidad de las pruebas que se realizan durante este proceso.

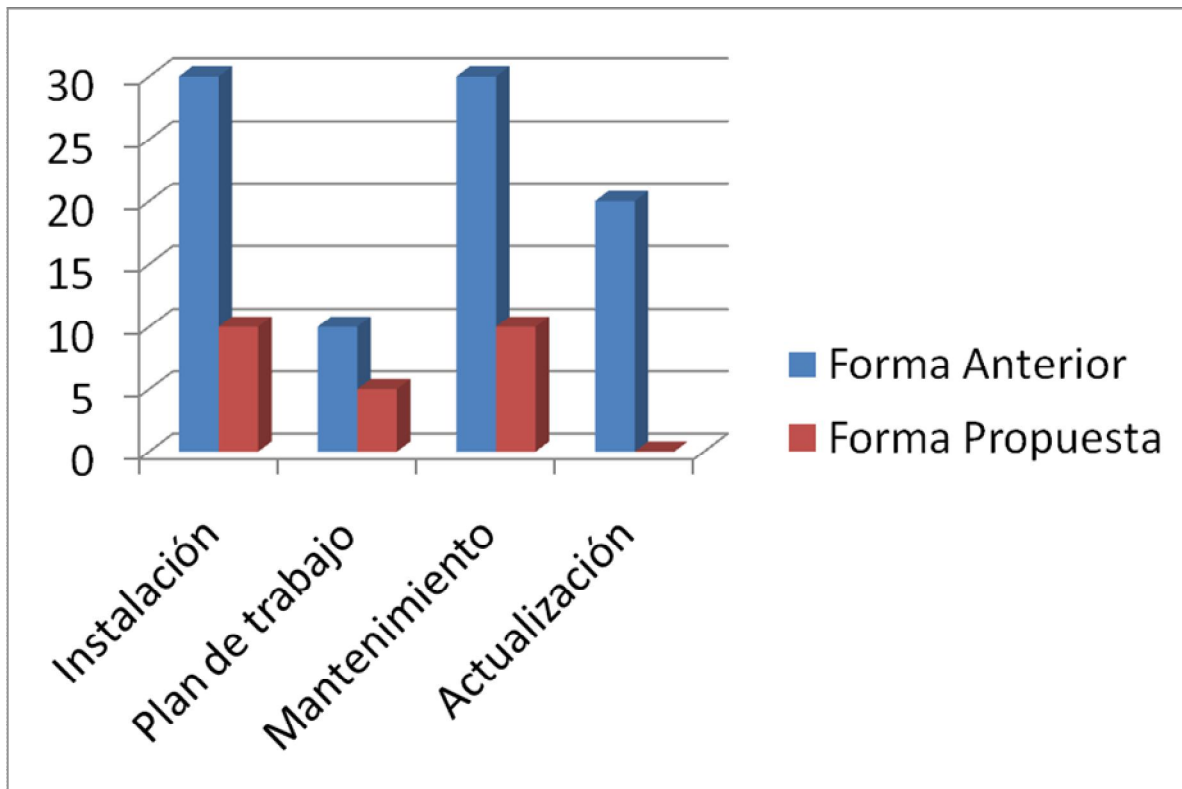
Valores de la variable (Forma Anterior):

- Instalación del software a los usuarios. 6 variables (30min.)
- Actualización del SW a los usuarios. 6 variables (20 min.)
- Mantenimiento del software. 3 variables (30 min.)
- Elaboración de un plan de trabajo. 1 variable (10 min.)

Valores de la variable (Forma Propuesta):

- Instalación del software a los usuarios. 6 variables (10 min.)
- Actualización del SW a los usuarios. 6 variables (0 min.)
- Mantenimiento del Sw. 3 variables (10 min.)
- Elaboración de un plan de trabajo. 1 variable (5 min.)

El gráfico que a continuación se describe, muestra el comportamiento de estas variables teniendo en cuenta las formas de realización de las actividades que componen el proceso.



Como se demuestra en el gráfico el sistema propuesto muestra una mayor eficacia y adaptabilidad a las nuevas circunstancias descritas anteriormente, motivo por el cual queda demostrado lo factible de la aplicación desarrollada.

4.2.3 Factibilidad operativa

La Factibilidad Operativa permite predecir, si se pondrá en marcha el sistema propuesto, aprovechando los beneficios que ofrece, a todos los usuarios involucrados con el mismo, ya sean los que interactúan en forma directa con este, como también que reciben información producida por el sistema. Por otra parte, el correcto funcionamiento del sistema en cuestión, siempre estará supeditado a la capacidad de los empleados encargados de dicha tarea.

La necesidad y deseo de un cambio en el sistema actual, expresada por los usuarios y el personal involucrado con el mismo, llevó a la aceptación de un nuevo sistema, que de una manera más sencilla y amigable, cubra todos sus

requerimientos, expectativas y proporciona, la información en forma segura y confiable. Basándose en las entrevistas y conversaciones sostenidas con el personal involucrado se demostró que estos no representan ninguna oposición al cambio, por lo que el sistema es factible operacionalmente.

Con la finalidad de garantizar el buen funcionamiento del sistema y que este impactará de forma positiva a los usuarios, el mismo fue desarrollado en forma estándar a los sistemas existentes en la institución, presentando una interfaz amigable al usuario, lo que se traduce en una herramienta de fácil manejo y comprensión, contando con los usuarios para cualquier modificación del sistema.

4.3 Conclusiones.

En este capítulo se realizó el estudio profundo del costo real en que se incurrió durante el diseño e implementación de la aplicación mediante la Metodología Costo Efectividad (Beneficios), se analizaron todos los factores directos, indirectos, externos e intangibles, así como se calculó el costo de ejecución del producto software mediante la ficha de costo, demostrándose la conveniencia de la elaboración del sistema.

CONCLUSIONES GENERALES

El sistema informático se desarrolló siguiendo la metodología XP. El sistema resultante está provisto de un ambiente cómodo, fácil de entender, que cumple los estándares de diseño y utiliza técnicas modernas de programación orientada a objetos, para esto se siguieron los siguientes pasos:

- Se elaboró el marco teórico metodológico que fundamenta la investigación, permitiendo el análisis del proceso de gestión de capacitación en la Empleni
- Se analizaron los sistemas informáticos existentes vinculados al objeto de estudio.
- Se realizó un estudio y selección de las tecnologías y herramientas a utilizar para el diseño e implementación de la aplicación.
- Se desarrolló el producto final, que consiste en el Diseño e Implementación de una aplicación Web para la gestión del proceso de capacitación en la EMPLENI.
- Se efectuó un estudio detallado de la factibilidad técnica, económica y operativa del software en cuestión, arrojando con este, resultados que muestran lo factible del proyecto.

Por lo antes expuesto se concluye que los objetivos propuestos en el presente proyecto han sido cumplidos satisfactoriamente.

RECOMENDACIONES

Con vista al desarrollo futuro de este proyecto se recomienda extender el software a cualquier entidad de nuestro país ya que la capacitación es un proceso desarrollado muchas entidades estatales de forma manual.

BIBLIOGRAFÍAS

Aguilar-Morales, J.E. (2010) Elaboración de programas de capacitación. Network de Psicología organizacional. México: Asociación Oaxaqueña de Psicología A.C.

BECK K. Extreme Programming Explained. Embrace Change [Libro]. - [s.l.]: Pearson Education, 1999.

CAVSI (2004) ¿Qué es un sistema gestor de Bases de Datos o SGBD? Disponible en: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.

Costo-Efectividad Disponible en:
www.crid.or.cr/digitalizacion/pdf/spa/doc9321/doc9321-02.pdf

Concepto de capacitación - Wikilearning Disponible
en: www.wikilearning.com/.../capacitacion...concepto_de_capacitacion

Concepto Importancia Capacitacion Disponible
en: www.mitecnologico.com/Main/ConceptoImportanciaCapacitacion

CAPITULO II Revisión de la Literatura 2.1. Antecedentes Históricos
Disponible
en: catarina.udlap.mx/u_dl_a/tales/documentos/lhr/.../capitulo2.pdf

Dondo Agustín (2005) PHP en castellano ¿Por qué elegir PHP? Disponible
en: <http://www.programacion.com/php/articulo/porquephp/>.

ECURED (La enciclopedia cubana en la red) Lenguajes de Programación/
Java Script Disponible en <http://www.ecured.cu/JavaScript>.

ECURED (La enciclopedia cubana en la red) Capacitación en América Latina y en Cuba Disponible en <http://www.ecured.cu/> Capacitación en América Latina y en Cuba.

EMS SQL Manager para MySQL Disponible en:
<http://sqlmanager.net/es/products/mysql/manager>

Graham Paul the Other Road Ahead [Libro]. - 2001.

Gestor de Base de Datos: MySQL. Gestor de Base de Datos: MySQL.
Disponible en: <http://Eaprende.com>.

kernelerror.net Arquitectura en tres capas. Disponible en:
<http://kernelerror.net/programacion/php/arquitectura-3-capas>

La Güeb de Joaquín UML -Tarjetas de CRC. Disponible
en:<http://jms32.eresmas.net/tacticos/UML/UML04/UML0402.html>

Masternewmedia.org Masternewmedia.org //Beneficios De Las Aplicaciones Basadas En Web Y El Anuncio De Microsoft De La Era "En Vivo". - 2005.

Microsoft SQL Server Disponible en:
http://es.wikipedia.org/wiki/Microsoft_SQL_Server.

Rrhh-web.com (La Web de los recursos humanos y el empleo)/Capacitación
Disponible en: <http://rrhh-web.com/capacitacion.html>

Rufino, A. (2007). "Lenguaje de programación: PHP." Disponible en:
<http://1sinfo.blogspot.com/2007/05/lenguaje-de-programacion-php.html>.

RUP. "Metodología de desarrollo software RUP." Disponible en:
http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational

Wikipedia.org Wikipedia, la enciclopedia libre (Interfaz de Usuario) Disponible en: http://es.wikipedia.org/wiki/Interfaz_de_usuario.

Zope. Disponible en: <http://es.wikipedia.org/wiki/Zope>.

SUBIRÓS MUÑOZ, Dariel Raúl. Desarrollo de una interfaz gráfica de usuario para el preprocesador meteorológico AERMET. Trabajo de Diploma. Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, 2009.

ZULUETA TORRES, Agustín. Modulo para la Extracción, Pre procesamiento, Descripción y Almacenaje en formato XML, de la información recuperada por el Sistema Automatizado de Información Virtual del ISMMM. Trabajo de Diploma. Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, 2010.

ARIAS RAMIREZ, Daimara. Sistema Automatizado para la Gestión de Información de la Maestría de Electromecánica. Trabajo de Diploma. Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, 2011.

ANEXO 1: HISTORIAS DE USUARIO

HU No.1: Gestión de perfiles

Historia de Usuario	
Número: 1	Usuario: Esp.B en Gestión de RR HH
Nombre de historia: Gestión de perfiles.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 3	Iteración Asignada: 1
Programador responsable: Leonart Urgellés Olivero.	
Descripción: El usuario debe insertar los datos de los Perfiles. Una vez añadidos, se podrá mostrar, actualizar, eliminar y buscar la información referente a ellos.	
Observaciones: Confirmado con el cliente.	

Tabla 2.4 HU No.1: Gestión de Perfiles

HU No.2: Gestión de competencias

Historia de Usuario	
Número: 2	Usuario: Esp.B en Gestión de RR HH
Nombre de historia: Gestión de competencias.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteración Asignada: 1
Programador responsable: Leonart Urgellés Olivero.	
Descripción: El usuario debe insertar los datos de las Competencias. Una vez añadidos, se podrá mostrar, actualizar, eliminar y buscar la información referente a ellos.	
Observaciones: Confirmado con el cliente.	

Tabla 2.5 HU No.1: Gestión de Competencias

HU No.3: Gestión de competencias del perfil

Historia de Usuario	
Número: 3	Usuario: Esp.B en Gestión de RR HH
Nombre de historia: Gestión de competencias del perfil.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 2	Iteración Asignada: 1
Programador responsable: Leonart Urgellés Olivero.	
Descripción: El usuario selecciona a que perfil le va a agregar la competencia y su importancia para el mismo. Una vez agregada, se podrá mostrar, eliminar y buscar la información referente a ellas.	
Observaciones: Confirmado con el cliente.	

Tabla 2.6 HU No.1: Gestión de competencias del perfil

HU No.4: Gestión de trabajadores

Historia de usuario	
Número: 4	Usuario: Capacitadora
Nombre: Gestión de trabajadores.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Leonart Urgellés Olivero.	
Descripción: El usuario debe insertar los datos de los trabajadores, estos una vez insertados se muestran en un listado. Una vez añadidos, se podrá mostrar, actualizar, eliminar y buscar la información referente a ellos.	
Observaciones: Confirmado con el cliente.	

Tabla 2.7 HU No.1: Gestión de trabajadores

HU No.5: Gestión de títulos

Historia de usuario	
Número: 5	Usuario: Capacitadora
Nombre: Gestión de títulos.	
Prioridad en el negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Leonart Urgellés Olivero.	
Descripción: El usuario selecciona el trabajador al cual le va asignar los títulos, estos una vez asignados se muestran en un listado y el usuario puede eliminar o modificar.	
Observaciones: Confirmado con el cliente.	

Tabla 2.8 HU No.1: Gestión de títulos

HU No.6: Gestión de evaluaciones

Historia de usuario	
Número: 6	Usuario: Comisión evaluadora
Nombre: Gestión de evaluaciones.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Leonart Urgellés Olivero.	
Descripción: El usuario agrega el trabajador y el año de la evaluación, una vez agregado debe seleccionar la chapa y la fecha para evaluar al trabajador, una vez seleccionados se podrá evaluar, ver sus evaluaciones y necesidades de capacitación, modificarlas y eliminarlas.	
Observaciones: Confirmado con el cliente.	

Tabla 2.9 HU No.1: Gestión de evaluaciones

HU No.7: Gestión de acciones

Historia de usuario	
Número: 6	Usuario: Comisión evaluadora
Nombre: Gestión de acciones.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Leonart Urgellés Olivero.	
Descripción: El usuario inserta las acciones, una vez insertadas podrá visualizarlas, modificarlas y eliminarlas, posteriormente si el trabajador está evaluado se le podrán asignar o eliminar dichas acciones.	
Observaciones: Confirmado con el cliente.	

Tabla 2.10 HU No.1: Gestión de acciones

ANEXO 2: TARJETAS CRC

Tarjeta CRC No.1 Gestión de perfiles

Nombre de la clase: Gestion _ perfiles	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Insertar datos de Perfiles.	Perfiles, Connections
Listar los Perfiles.	Perfiles, Connections
Modificar los Perfiles.	Perfiles, Connections
Eliminar los Perfiles.	Perfiles, Connections
Ver generalidades del Perfil.	Perfiles, Connections

Tabla 2.13 Gestión de perfiles

Tarjeta CRC No.2 Gestión de competencias

Nombre de la clase: Gestion _ competencias	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Insertar datos de Competencias.	Competencias, Connections
Listar las Competencias.	Competencias, Connections
Modificar las Competencias.	Competencias, Connections
Eliminar las Competencias.	Competencias, Connections

Tabla 2.14 Gestión de competencias

Tarjeta CRC No.3 Gestión de competencias del perfil

Nombre de la clase: Gestion _ competencias _ perfil	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Agregar competencias a un perfil.	Perfiles, Competencias, Connections
Eliminar las competencias de un perfil.	Perfiles, Competencias, Connections
Asignar importancia a competencias de un perfil.	Perfiles, Competencias, Connections

Tabla 2.15 Gestión de Competencias del Perfil

Tarjeta CRC No.4 Gestión de trabajadores

Nombre de la clase: Gestion _ trabajadores	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Insertar un trabajador.	Trabajadores, Connections
Listar trabajadores.	Trabajadores, Connections
Eliminar trabajadores.	Trabajadores, Connections
Modificar datos de trabajadores.	Trabajadores, Connections

Tabla 2.16 Gestión de trabajadores

Tarjeta CRC No.5 Gestión de títulos

Nombre de la clase: Gestion _titulos	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Asignar títulos a un trabajador.	Titulos, Trabajadores, Connections
Ver títulos de un trabajador.	Titulos, Trabajadores, Connections
Eliminar títulos de un trabajador.	Titulos, Trabajadores, Connections
Modificar datos de trabajadores.	Titulos, Trabajadores, Connections

Tabla 2.17 Gestión de títulos

Tarjeta CRC No.6 Gestión de evaluaciones

Nombre de la clase: Gestion _evaluaciones	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Insertar evaluación de competencias.	Perfiles, Competencias, Trabajadores, Connections
Mostrar evaluación de competencias.	Perfiles, Competencias, Trabajadores, Connections
Modificar evaluación de competencias.	Perfiles, Competencias, Trabajadores, Connections
Determinar evaluación final y necesidad de capacitación.	Perfiles, Competencias, Trabajadores, Connections
Mostrar evaluación final y necesidad de capacitación.	Perfiles, Competencias, Trabajadores, Connections

Tabla 2.18 Gestión de evaluaciones

Tarjeta CRC No.7 Gestión de acciones

Nombre de la clase: Gestion_acciones	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Insertar acciones.	Acciones, Trabajadores, Evaluaciones, Connections
Mostrar acciones.	Acciones, Trabajadores, Evaluaciones, Connections
Modificar acciones.	Acciones, Trabajadores, Evaluaciones, Connections
Eliminar acciones.	Acciones, Trabajadores, Evaluaciones, Connections
Asignar acciones a una evaluación de un trabajador.	Acciones, Trabajadores, Evaluaciones, Connections
Eliminar acciones de un trabajador.	Acciones, Trabajadores, Evaluaciones, Connections

Tabla 2.19 Gestión de acciones

ANEXO 3: INTERFACES DE USUARIO

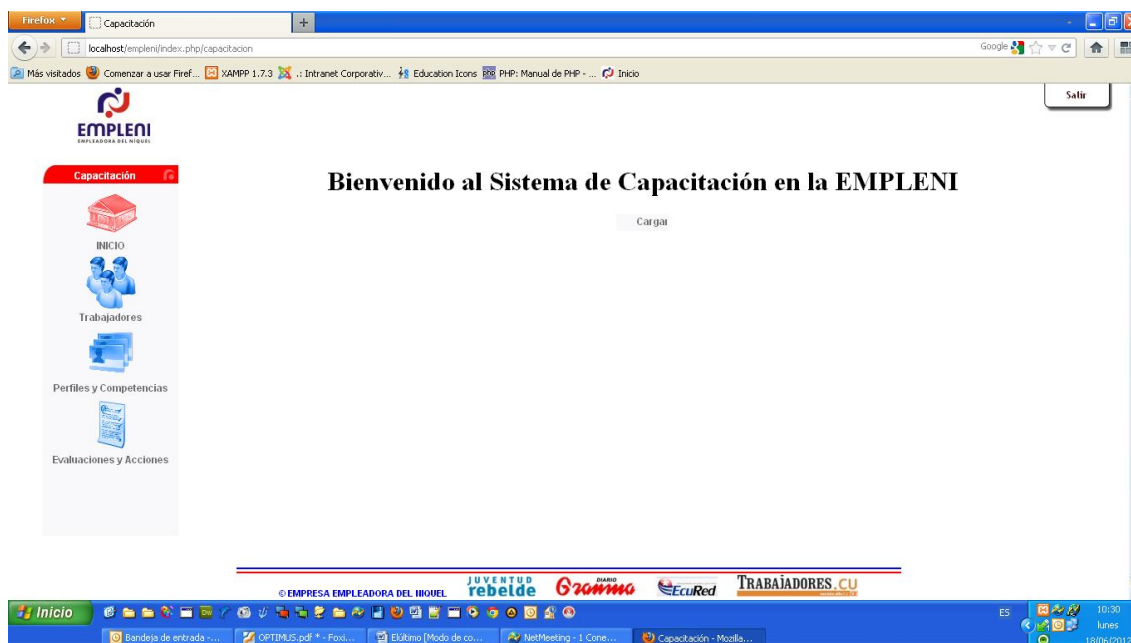


Figura 2.1 Portada del Sistema.

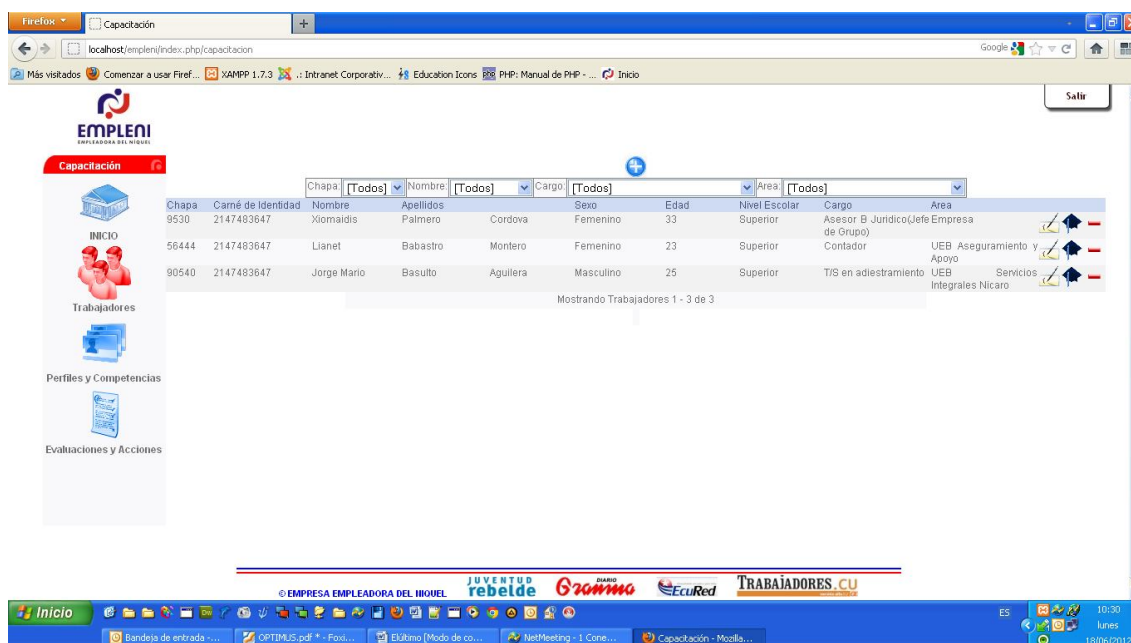


Figura 2.2 Lista de trabajadores.

ANEXO 3: INTERFACES DE USUARIO

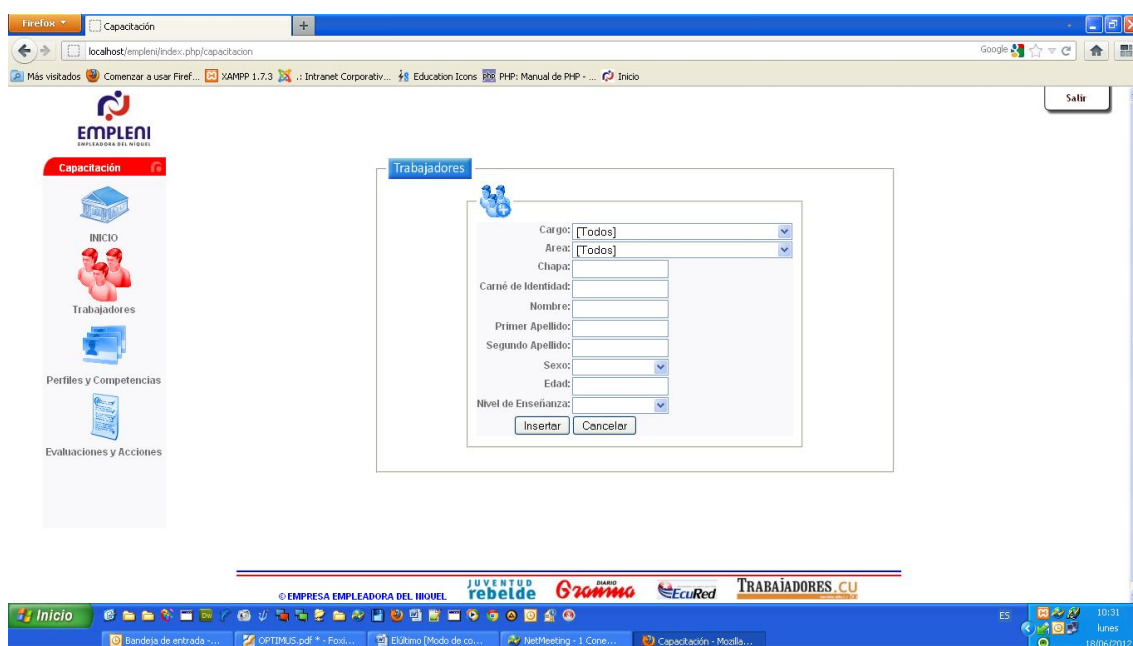


Figura 2.3 Insertar Trabajadores.

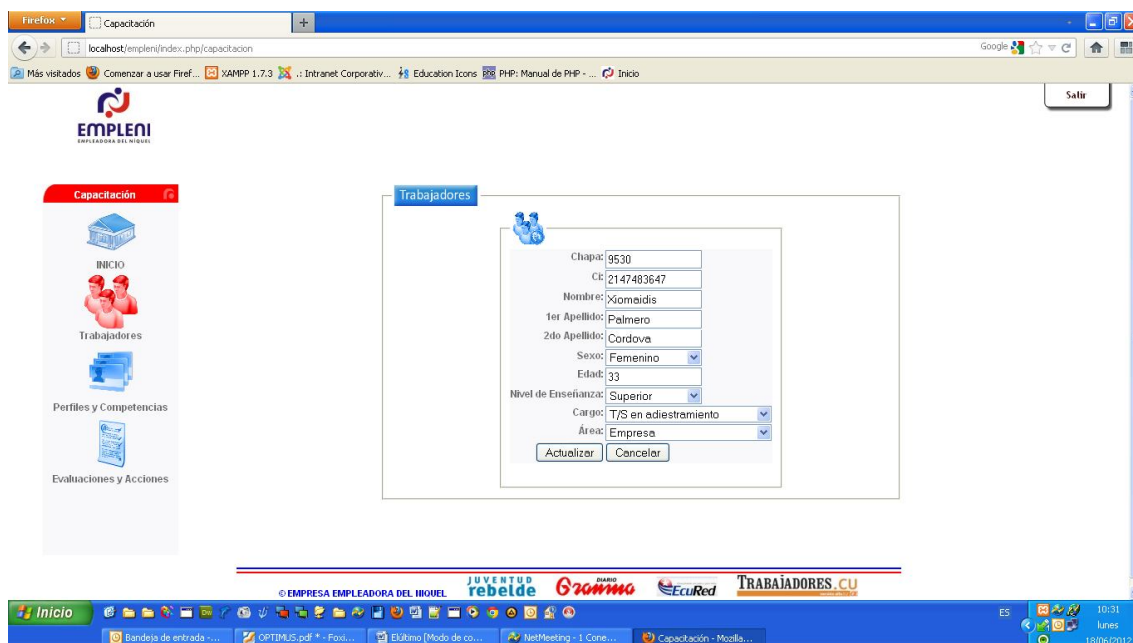


Figura 2.4 Modificar Trabajadores.

ANEXO 3: INTERFACES DE USUARIO

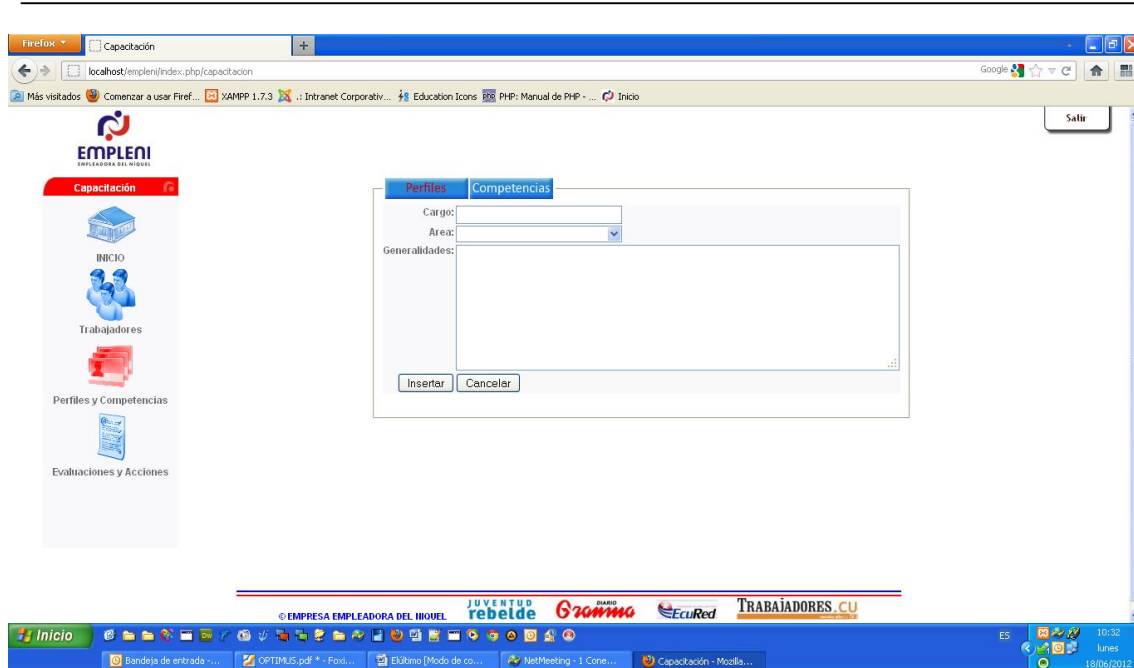


Figura 2.5 Insertar Perfiles.

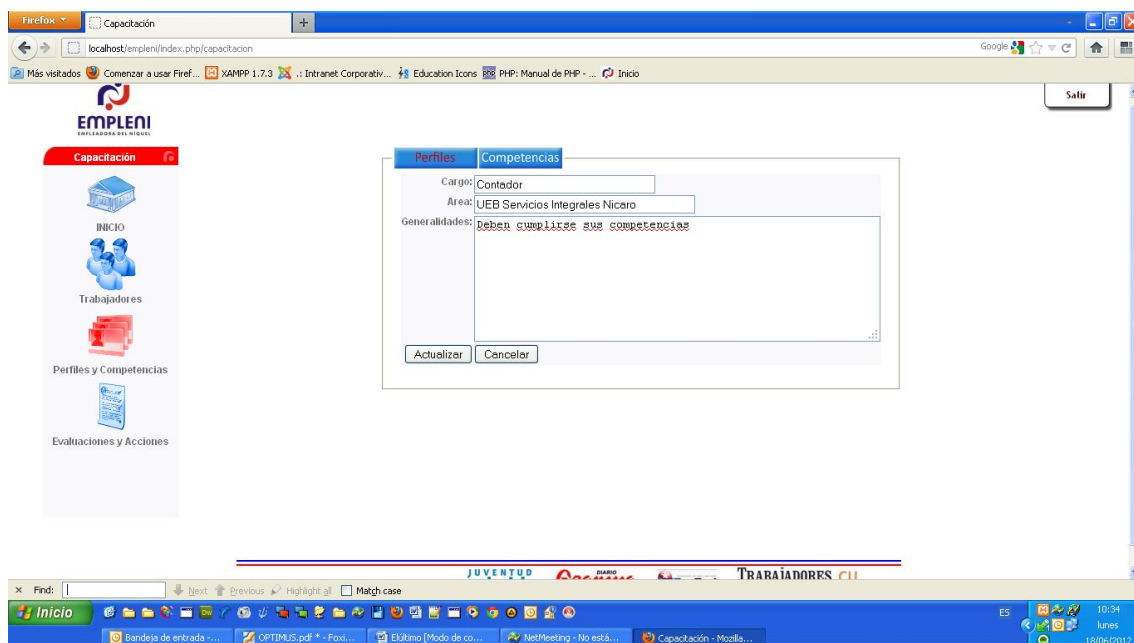


Figura 2.6 Modificar Perfiles.

ANEXO 3: INTERFACES DE USUARIO

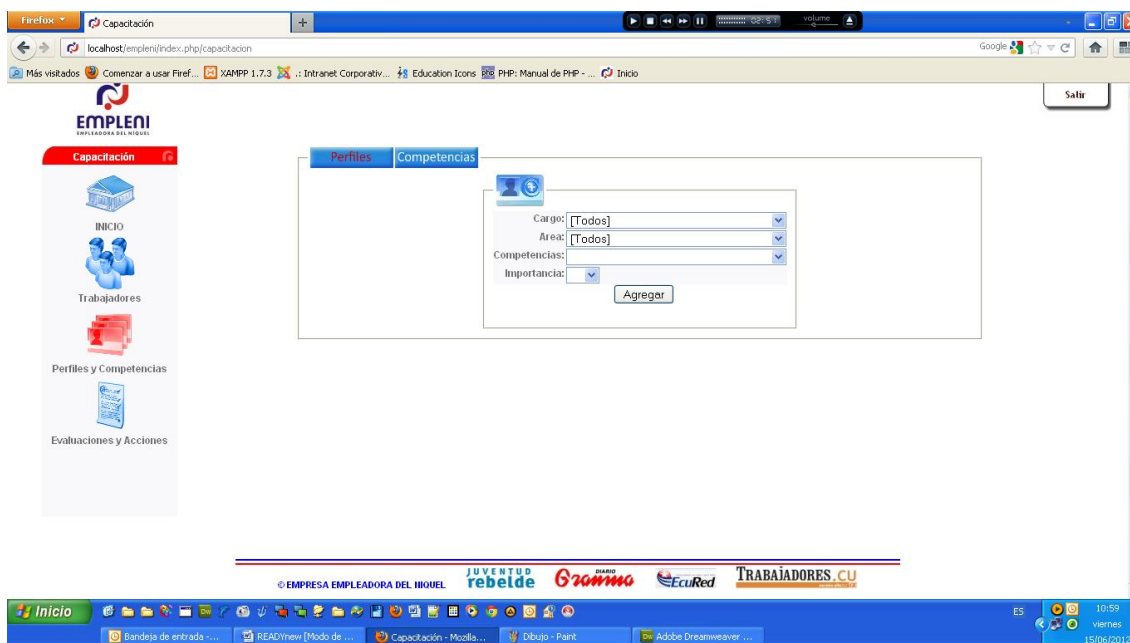


Figura 2.7 Agregar competencias a un perfil.

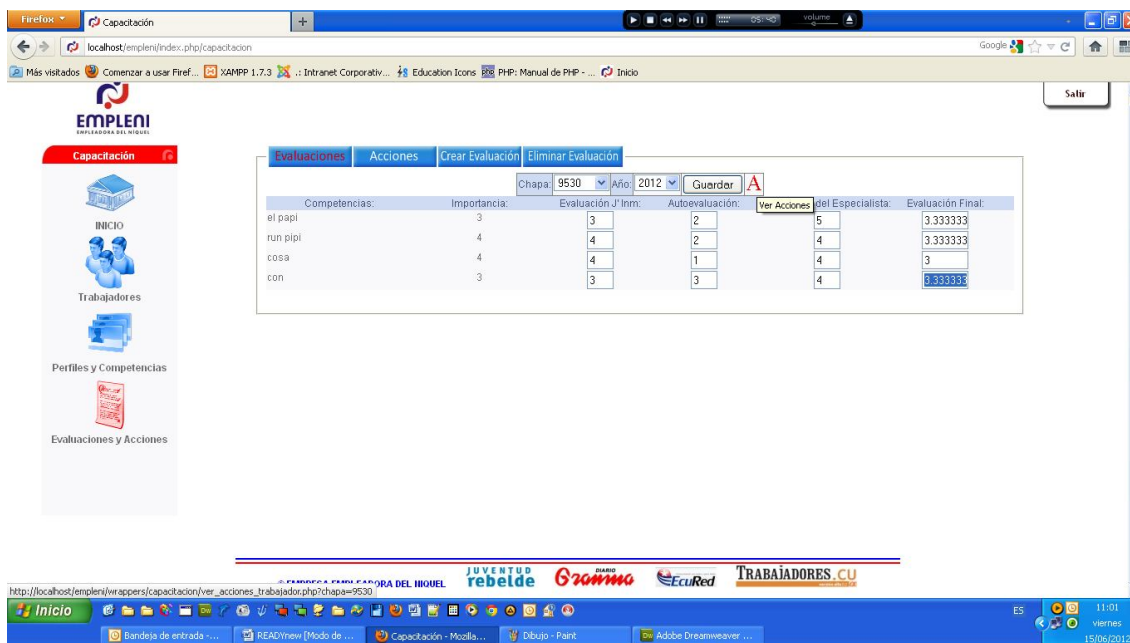


Figura 2.8 Evaluaciones de un trabajador.

ANEXO 4: TARJETAS DE TAREAS

Tarjeta de tarea No. 1: Insertar perfil

Tarea de ingeniería	
Número tarea: 1	Número historia: 1
Nombre tarea: Insertar perfil.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 20/02/2012	Fecha fin: 22/02/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea facilita introducir los datos de un perfil.	

Tabla 3.3 Tarjeta de tarea No. 1

Tarjeta de tarea No. 2: Listar perfil

Tarea de ingeniería	
Número tarea: 2	Número historia: 1
Nombre tarea: Listar perfil.	
Tipo de tarea : Diseño	Puntos estimados: 0.2
Fecha inicio: 23/02/2012	Fecha fin: 24/02/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite visualizar los datos de un perfil.	

Tabla 3.4 Tarjeta de tarea No. 2

Tarjeta de tarea No. 3: Modificar perfil

Tarea de ingeniería	
Número tarea: 3	Número historia: 1
Nombre tarea: Modificar perfil.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 27/02/2012	Fecha fin: 29/02/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite modificar los datos de un perfil.	

Tabla 3.5 Tarjeta de tarea No. 3

Tarjeta de tarea No. 4: Eliminar perfil

Tarea de ingeniería	
Número tarea: 4	Número historia: 1
Nombre tarea: Eliminar perfil.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 1/03/2012	Fecha fin: 5/03/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite eliminar los datos de un perfil.	

Tabla 3.6 Tarjeta de tarea No. 4

Tarjeta de tarea No. 5: Ver generalidades del perfil

Tarea de ingeniería	
Número tarea: 5	Número historia: 1
Nombre tarea: Ver generalidades del perfil.	
Tipo de tarea : Diseño	Puntos estimados: 0.2
Fecha inicio: 6/03/2012	Fecha fin:7/03/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite ver los datos de un perfil.	

Tabla 3.7 Tarjeta de tarea No. 5

Tarjeta de tarea No. 6: Insertar competencias

Tarea de ingeniería	
Número tarea: 6	Número historia: 2
Nombre tarea: Insertar competencias.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 8/03/2012	Fecha fin:10/03/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite insertar los datos de una competencia.	

Tabla 3.8 Tarjeta de tarea No. 6

Tarjeta de tarea No. 7: Listar competencias

Tarea de ingeniería	
Número tarea: 7	Número historia: 2
Nombre tarea: Listar competencias.	
Tipo de tarea : Diseño	Puntos estimados: 2
Fecha inicio: 12/03/2012	Fecha fin: 13/03/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite visualizar los datos de una competencia.	

Tabla 3.9 Tarjeta de tarea No. 7

Tarjeta de tarea No. 8: Eliminar competencias

Tarea de ingeniería	
Número tarea: 8	Número historia: 2
Nombre tarea: Eliminar competencias.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 14/03/2012	Fecha fin: 16/03/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite eliminar los datos de una competencia.	

Tabla 3.10 Tarjeta de tarea No. 8

ANEXO 4: TARJETAS DE TAREAS

Tarjeta de tarea No.9: Modificar competencias

Tarea de ingeniería	
Número tarea: 9	Número historia: 2
Nombre tarea: Modificar competencias.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 19/03/2012	Fecha fin:21/03/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite Modificar los datos de una competencia una vez que son insertados.	

Tabla 3.11 Tarjeta de tarea No. 9

Tarjeta de tarea No.10: Agregar competencias a un perfil

Tarea de ingeniería	
Número tarea: 10	Número historia: 3
Nombre tarea: Agregar competencias a un perfil.	
.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 22/03/2012	Fecha fin:24/03/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite Agregar una competencia a un perfil una vez insertado el perfil y la competencia.	

Tabla 3.12 Tarjeta de tarea No. 10

Tarjeta de tarea No.11: Eliminar competencias de un perfil

Tarea de ingeniería	
Número tarea: 11	Número historia: 3
Nombre tarea: Eliminar competencias de un perfil.	
.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 26/02/2012	Fecha fin:28/03/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite Eliminar una competencia de un perfil una vez insertado el perfil y asignada la competencia a ese perfil.	

Tabla 3.13 Tarjeta de tarea No. 11

Tarjeta de tarea No.12: Asignar importancia a competencias de un perfil

Tarea de ingeniería	
Número tarea: 12	Número historia: 3
Nombre tarea: Asignar importancia a competencias de un perfil.	
.	
Tipo de tarea : Desarrollo	Puntos estimados: 0.2
Fecha inicio: 29/03/2012	Fecha fin:30/03/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite asignar importancia a una competencia de un perfil específico.	

Tabla 3.14 Tarjeta de tarea No. 12

Tarjeta de tarea No.13: Insertar trabajador

Tarea de ingeniería	
Número tarea: 13	Número historia: 4
Nombre tarea: Insertar trabajador. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 2/04/2012	Fecha fin:4/04/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite insertar un trabajador en la bd y a la vez asignarle un perfil.	

Tabla 3.15 Tarjeta de tarea No. 13

Tarjeta de tarea No.14: Listar trabajador

Tarea de ingeniería	
Número tarea: 14	Número historia: 4
Nombre tarea: Insertar trabajador. .	
Tipo de tarea : Diseño	Puntos estimados: 0.2
Fecha inicio: 5/04/2012	Fecha fin:6/04/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite visualizar los datos correspondientes a un trabajador.	

Tabla 3.16 Tarjeta de tarea No. 14

Tarjeta de tarea No.15: Eliminar trabajador

Tarea de ingeniería	
Número tarea: 15	Número historia: 4
Nombre tarea: Eliminar trabajador. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 9/04/2012	Fecha fin:11/04/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite eliminar los datos correspondientes a un trabajador.	

Tabla 3.17 Tarjeta de tarea No. 15

Tarjeta de tarea No.16: Modificar Trabajador

Tarea de ingeniería	
Número tarea: 16	Número historia: 4
Nombre tarea: Modificar trabajador. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 12/04/2012	Fecha fin:14/04/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite modificar los datos correspondientes a un trabajador una vez insertado el trabajador.	

Tabla 3.18 Tarjeta de tarea No. 16

ANEXO 4: TARJETAS DE TAREAS

Tarjeta de tarea No.17: Asignar títulos a un trabajador

Tarea de ingeniería	
Número tarea: 17	Número historia: 5
Nombre tarea: Asignar títulos a un trabajador. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 16/04/2012	Fecha fin:18/04/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite asignar uno o varios títulos a un trabajador una vez insertado el trabajador.	

Tabla 3.19 Tarjeta de tarea No. 17

Tarjeta de tarea No.18: Ver títulos de un trabajador

Tarea de ingeniería	
Número tarea: 18	Número historia: 5
Nombre tarea: Ver títulos de un trabajador. .	
Tipo de tarea : Diseño	Puntos estimados: 0.2
Fecha inicio: 19/04/2012	Fecha fin:20/04/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite visualizar el o los títulos de un trabajador una vez insertado el trabajador y asignado su título.	

Tabla 3.20 Tarjeta de tarea No. 18

Tarjeta de tarea No.19: Eliminar títulos de un trabajador

Tarea de ingeniería

ANEXO 4: TARJETAS DE TAREAS

Número tarea: 19	Número historia: 5
Nombre tarea: Eliminar títulos de un trabajador. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.2
Fecha inicio: 23/04/2012	Fecha fin: 24/04/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite eliminar el o los títulos de un trabajador una vez insertado el trabajador y asignado su título.	

Tabla 3.21 Tarjeta de tarea No. 19

Tarjeta de tarea No.20: Modificar Títulos de un Trabajador

Tarea de ingeniería	
Número tarea: 20	Número historia: 5
Nombre tarea: Modificar títulos de un trabajador. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.2
Fecha inicio: 25/04/2012	Fecha fin: 27/04/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite modificar el o los títulos de un trabajador una vez asignado su título.	

Tabla 3.22 Tarjeta de tarea No. 20

Tarjeta de tarea No.21: Insertar evaluación de competencias

ANEXO 4: TARJETAS DE TAREAS

Tarea de ingeniería	
Número tarea: 21	Número historia: 6
Nombre tarea: Insertar evaluación de competencias. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 2/05/2012	Fecha fin:4/05/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite evaluar las competencias de un trabajador una vez asignadas las competencias a su perfil, a cada competencia se realizan tres evaluaciones del (Jefe inmediato, auto evaluación, evaluación del especialista).	

Tabla 3.23 Tarjeta de tarea No. 21

Tarjeta de tarea No.22: Mostrar evaluaciones de competencias

Tarea de ingeniería	
Número tarea: 22	Número historia: 6
Nombre tarea: Mostrar evaluaciones de competencias. .	
Tipo de tarea : Diseño	Puntos estimados: 0.2
Fecha inicio: 7/05/2012	Fecha fin:8/05/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite mostrar las evaluaciones de las competencias de un trabajador una vez evaluado este.	

Tabla 3.24 Tarjeta de tarea No. 22

Tarjeta de tarea No.23: Modificar evaluaciones de competencias

ANEXO 4: TARJETAS DE TAREAS

Tarea de ingeniería	
Número tarea: 23	Número historia: 6
Nombre tarea: Modificar evaluaciones de competencias. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 9/05/2012	Fecha fin: 11/05/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite modificar las evaluaciones en caso de que existan cambios o se pospongan las evaluaciones.	

Tabla 3.25 Tarjeta de tarea No. 23

Tarjeta de tarea No.24: Determinar evaluación final y necesidad de capacitación

Tarea de ingeniería	
Número tarea: 24	Número historia: 6
Nombre tarea: Determinar evaluación final y necesidad de capacitación. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 14/05/2012	Fecha fin: 16/05/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite calcular la evaluación final y la necesidad de capacitación de los trabajadores a partir de las evaluaciones realizadas anteriormente.	

Tabla 3.26 Tarjeta de tarea No. 24

Tarjeta de tarea No.25: Mostrar evaluación final y necesidad de capacitación

ANEXO 4: TARJETAS DE TAREAS

Tarea de ingeniería	
Número tarea: 25	Número historia: 6
Nombre tarea: Determinar evaluación final y necesidad de capacitación. .	
Tipo de tarea : Diseño	Puntos estimados: 0.2
Fecha inicio: 17/05/2012	Fecha fin: 18/05/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite visualizar la evaluación final de los trabajadores y necesidad de capacitación, para después determinar si se debe aplicar alguna acción en particular para esa competencia en correspondencia con su importancia para este perfil.	

Tabla 3.27 Tarjeta de tarea No. 25

Tarjeta de tarea No.26: Insertar acciones

Tarea de ingeniería	
Número tarea: 26	Número historia: 7
Nombre tarea: Insertar acciones. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 21/05/2012	Fecha fin: 23/05/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite insertar una acción para después esta ser aplicada a un trabajador según su evaluación final y la importancia de esa competencia para el perfil.	

Tabla 3.28 Tarjeta de tarea No. 26

Tarjeta de tarea No.27: Mostrar acciones

Tarea de ingeniería	
Número tarea: 27	Número historia: 7
Nombre tarea: Mostrar acciones. .	
Tipo de tarea : Diseño	Puntos estimados: 0.2
Fecha inicio: 24/05/2012	Fecha fin: 25/05/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite mostrar una acción una vez que esta fue insertada.	

Tabla 3.29 Tarjeta de tarea No. 27

Tarjeta de tarea No.28: Modificar acciones

Tarea de ingeniería	
Número tarea: 28	Número historia: 7
Nombre tarea: Modificar acciones. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 28/05/2012	Fecha fin: 30/05/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite modificar una acción una vez que esta fue insertada.	

Tabla 3.30 Tarjeta de tarea No. 28

Tarjeta de tarea No.29: Eliminar acciones

Tarea de ingeniería	
Número tarea: 29	Número historia: 7
Nombre tarea: Eliminar acciones. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.2
Fecha inicio: 31/05/2012	Fecha fin:1/06/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite eliminar una acción una vez que esta fue insertada.	

Tabla 3.31 Tarjeta de tarea No. 29

Tarjeta de tarea No.30: Asignar acciones a un trabajador

Tarea de ingeniería	
Número tarea: 30	Número historia: 7
Nombre tarea: Asignar acciones a un trabajador. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.3
Fecha inicio: 4/06/2012	Fecha fin:6/06/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite asignar una acción a un trabajador una vez que esta fue insertada teniendo en cuenta su evaluación final y la importancia de la competencia para su perfil.	

Tabla 3.32 Tarjeta de tarea No. 30

ANEXO 4: TARJETAS DE TAREAS

Tarjeta de tarea No.31: Eliminar acciones de un trabajador

Tarea de ingeniería	
Número tarea: 31	Número historia: 7
Nombre tarea: Eliminar acciones de un trabajador. .	
Tipo de tarea : Desarrollo	Puntos estimados: 0.2
Fecha inicio: 7/06/2012	Fecha fin:8/06/2012
Programador responsable: Leonart Urgellés Olivero.	
Descripción: Esta tarea permite eliminar una acción de un trabajador una vez que esta le fue asignada.	

Tabla 3.33 Tarjeta de tarea No. 31

ANEXO 5: PRUEBAS DE ACEPTACION

Pruebas de aceptación para la HU No.1: Gestión de perfiles

Prueba de aceptación
HU: 1
Nombre: Gestión de perfiles.
Descripción: La especialista debe suministrarle al sistema los datos de los perfiles para luego mostrarlos, modificarlos, eliminarlos y ver sus generalidades.
Condiciones de ejecución: El usuario debe introducir los datos de los perfiles.
Entrada/Pasos ejecución: • Insertar el perfil • Ver el listado de perfiles • Modificar un perfil • Eliminar un perfil • Ver generalidades de un perfil
Resultado Esperado: Gestionar correctamente el perfil
Evaluación de la prueba: Aceptada.

Tabla 3.35 PA: Prueba para comprobar la gestión de perfiles.

Pruebas de aceptación para la HU No.2: Gestión de competencias

Prueba de aceptación
HU: 2
Nombre: Gestión de competencias.
Descripción: La especialista debe suministrarle al sistema los datos de las competencias para luego mostrarlos, modificarlos, eliminarlos.
Condiciones de ejecución: El usuario debe introducir los datos de las competencias.
Entrada/Pasos ejecución: • Insertar la competencia • Ver el listado de competencias • Modificar la competencia deseada • Eliminar una competencia
Resultado Esperado: Gestionar Correctamente la competencia
Evaluación de la prueba: Aceptada.

Tabla 3.36 PA: Prueba para comprobar la gestión de competencias.

Pruebas de aceptación para la HU No.3: Gestión de competencias del perfil

Prueba de aceptación
HU: 3
Nombre: Gestión de competencias perfil.
Descripción: La especialista debe agregar la competencia a un perfil para luego esta poder ser eliminada del perfil y tener asignada una importancia.
Condiciones de ejecución: El usuario debe asignar la competencia y su importancia a un perfil.
Entrada/Pasos ejecución: • Agregar la competencia • Asignar importancia a dicha competencia • Eliminar competencia de un perfil
Resultado Esperado: Gestionar Correctamente la competencia
Evaluación de la prueba: Aceptada.

Tabla 3.37 PA: Prueba para comprobar la gestión de competencias del perfil.

Pruebas de aceptación para la HU No.4: gestión de trabajadores

Prueba de aceptación
HU: 4
Nombre: Gestión de trabajadores.
Descripción: La especialista debe agregar los datos de un trabajador para luego estos poder ser mostrados en un listado, eliminados, modificados.
Condiciones de ejecución: El usuario debe insertar un trabajador.
Entrada/Pasos ejecución: • Insertar trabajador • Ver los datos del trabajador • Modificar los datos de dicho trabajador • Eliminar el trabajador
Resultado Esperado: Gestionar Correctamente el trabajador
Evaluación de la prueba: Aceptada.

Tabla 3.38 PA: Prueba para comprobar la gestión de trabajadores.

Pruebas de aceptación para la HU No.5: Gestión de títulos

Prueba de aceptación
HU: 5
Nombre: Gestión de títulos.
Descripción: La especialista debe asignar un titulo a los trabajadores, verlos en un listado, modificarlos y luego eliminarlos.
Condiciones de ejecución: El usuario debe insertar un título para luego realizar las operaciones.
Entrada/Pasos ejecución: • Asignar titulo a un trabajador • Ver los títulos del trabajador • Modificar los títulos de dicho trabajador • Eliminar los títulos del trabajador
Resultado Esperado: Gestionar Correctamente el título
Evaluación de la prueba: Aceptada.

Tabla 3.39 PA: Prueba para comprobar la gestión de títulos.

Pruebas de aceptación para la HU No.6: Gestión de evaluaciones

Prueba de aceptación
HU: 6
Nombre: Gestión de evaluaciones.
Descripción: El representante de la comision evaluadora designado debe insertar las evaluaciones, para luego estas ser visualizadas, modificadas y posteriormente ver la evaluación final y la necesidad de capacitacion.
Condiciones de ejecución: El usuario debe insertar las evaluaciones para luego realizar las operaciones.
Entrada/Pasos ejecución: • Insertar evaluación • Ver las evaluaciones y las necesidades de capacitación • Modificar las evaluaciones
Resultado Esperado: Gestionar Correctamente las evaluaciones
Evaluación de la prueba: Aceptada.

Tabla 3.40 PA: Prueba para comprobar la gestión de evaluaciones.

Pruebas de aceptación para la HU No.7: Gestión de acciones

Prueba de aceptación
HU: 7
Nombre: Gestión de acciones.
Descripción: la comisión evaluadora debe insertar una acción para luego visualizarla, modificarla, eliminarla y también puede ser asignada a un trabajador.
Condiciones de ejecución: El usuario debe insertar las acciones para luego realizar las operaciones.
Entrada/Pasos ejecución: • Insertar acción • Ver las acciones • Modificarlas • Eliminarlas • Asignarlas a trabajadores • Eliminar Acción de trabajadores
Resultado Esperado: Gestionar Correctamente las acciones
Evaluación de la prueba: Aceptada

Tabla 3.41 PA: Prueba para comprobar la gestión de acciones.