

**Sistema de gestión para el Departamento
de Recursos Humanos de la Dirección de
Producción Audiovisual del centro**

FORTES

Trabajo para optar por el Título de Ingeniero en Ciencias
Informáticas

Autores:

Idalmis Rodríguez Ramírez

Alejandro Luis Hernández Armas

Tutores:

Ing. Jorge Antonio Díaz Gutiérrez

Ing. Arlan Gálvez Alonso

Ciudad de La Habana, Julio de 2011.

Declaración de Autoría

Declaramos que somos los únicos autoras del presente Trabajo de Diploma que lleva como título: Sistema de gestión para el Departamento de Recursos Humanos de la Dirección de Producción Audiovisual del centro FORTES y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

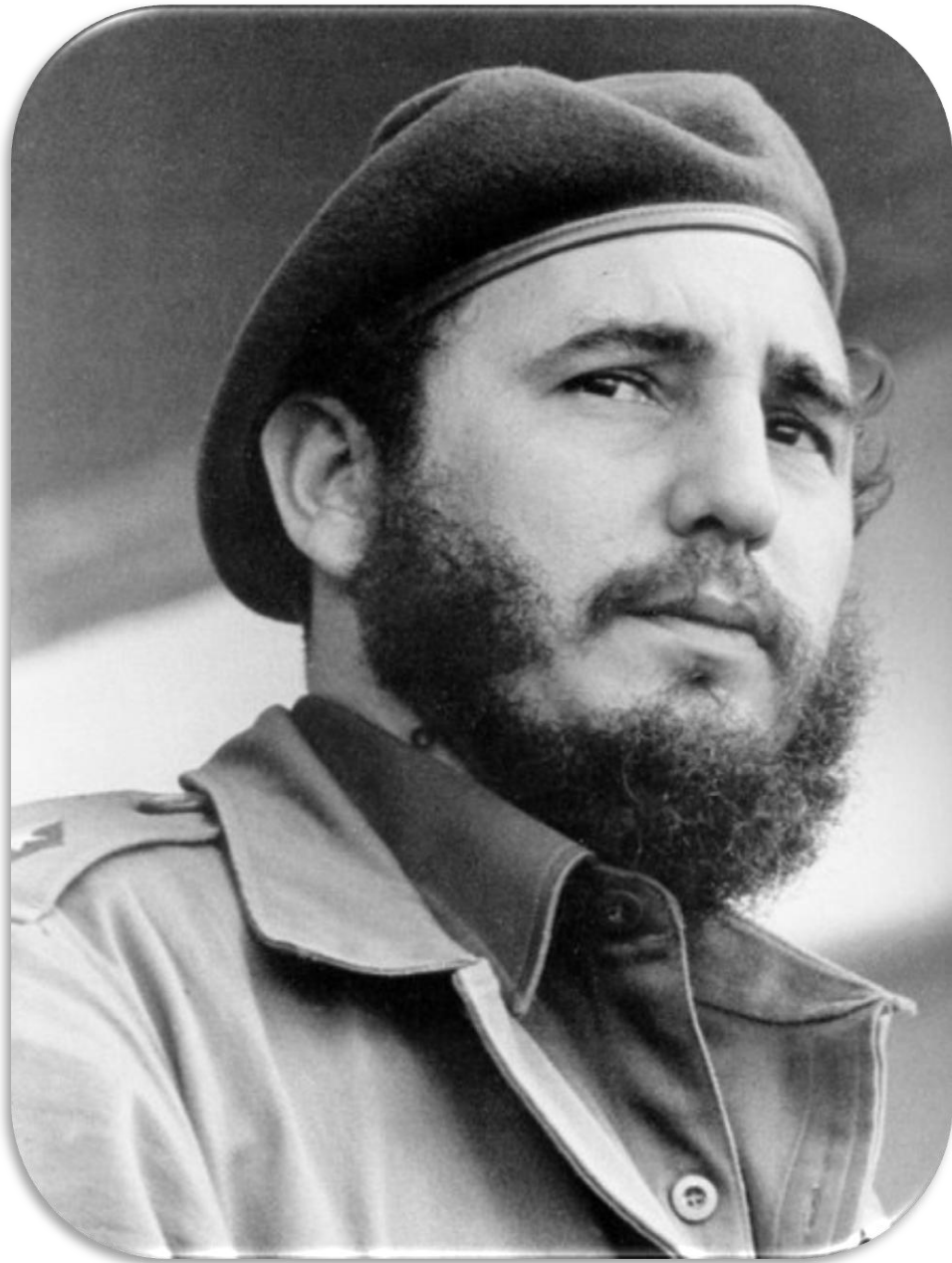
Para que así conste firmamos la presente a los _____ días del mes de _____ del año _____.

Firma del Autor
Idalmis Rodríguez Ramírez

Firma del Autor
Alejandro Luis Hernández Armas

Firma del Tutor
Ing. Jorge Antonio Díaz Gutiérrez

Firma del Tutor
Ing. Arlan Gálvez Alonso



“El futuro de Cuba tiene que ser necesariamente un futuro de hombres de ciencia, de hombres de pensamiento.” Fidel Castro

Dedicatoria

A mi familia, en especial a mis abuelos, mis padres y mi hermanito por tanto amor, esfuerzo, comprensión, por ser la razón de mi vida...y ayudarme a ser mejor persona cada día.

Idalmis Rodríguez Ramírez

A Dios, por ser mi cómplice, mi amuleto en las horas difíciles, por regalarme la FE de cada día. A mis padres que tanto amo, a ellos que adivinan hasta el último de mis sueños y hacen que el tiempo no se resista bajo sus pies, a veces cansados, a ellos que con su ejemplo han sabido guiarme por el camino del bien. A mi Tata querida, Arlenys, la mejor y más linda de las hermanas, sé que desde la distancia que el deber impone estás hoy muy cerca de mí. A mi abuela Estrella y a Ñañy por sus oraciones. A mi tía preciosa María, por tanto cariño. A mis amigos y hermanos de todos los tiempos: Andrisael, Osvaldo, Raydel y Julio.

Alejandro Luis Hernández Armas

Agradecimientos

A mi familia especialmente a mis padres, abuelos y hermanito por su apoyo incondicional, por darme las fuerzas necesarias cuando he creído haber perdido una batalla, por confiar siempre en mí, por ser mi principal fuente de inspiración para avanzar en la vida, por el esfuerzo constante y una vida de sacrificios...por ellos es que he podido convertirme en la mujer que realmente soy... no tengo palabras para expresar el gran amor, respeto y admiración que siento por ellos.



A mi papito por ser siempre mi caballero perfecto, por sus palabras de aliento en mis momentos más difíciles, por instruirme en la vida, por el valor y el coraje que ha tenido para levantarse ante cualquier adversidad, por sus consejos, por cumplir con sus deberes y nunca fallarme, porque sé que siempre puedo contar contigo, hombres como tú hay pocos, gracias por ser un padre ideal.

A mi mamita por el gran amor y devoción que tienes a tus hijos por el apoyo ilimitado e incondicional que siempre me has dado, por haberme dado la vida, por haberme formado como una mujer de bien, por compartir conmigo todos esos valores que te hacen la mujer más especial del mundo: tu bondad, perdón, honestidad, perseverancia, consideración y paciencia, por escuchar que la primera palabra que dije fue: papá... (Perdóname)... eres la persona más dulce que tengo en el mundo te amo mucho.

A la profesora Surelys por todo el apoyo, preocupación y dedicación que ha tenido con nosotros, ha sido como :(un ángel de la guarda), no tenemos como agradecerle todo lo que ha hecho por nosotros. Gracias

A mis amigos de primer año del grupo 8106 a Yusdel, Papa, Osvaldo, Chino, Aldo, Leisel, Cindy, Dania, Mairín, Yarenmy, a una personita que es muy especial para mí, Elizabeth sabes lo mucho que significas para mí, gracias por estar ahí siempre y enseñarme el verdadero sentido de la amistad, te quiero muchísimo.

A los amigos del grupo 8307 por compartir momentos de alegrías juntos y porque se que el cariño que siento por ustedes es recíproco agradezco en especial a Liset y el Yero, Clary, Jessy, Lia y Adrián, Zuly y Rogney, Anett, Julio, Agustín y Mily, Ancel, Leo, Yane, Flako y un agradecimiento especial a la flaquita mas linda del mundo: a Anita por todo el apoyo que siempre me brindaste y en todo lo que me ayudaste, por ser mi amiga, no tengo como agradecerte lo que haces por mi y por Alex, un millón de gracias.

A mi compañero de tesis Alex por ser tan buen amigo y ser tan lindo siempre conmigo, por salir de esta los dos juntitos no hubiera querido tener otro compañero de tesis que no fueras tú... te quiero Hellscream.

Un millón de palabras o lágrimas no pueden agradecerles a todos. Cultivar verdaderos amigos requiere dedicación y tiempo. No puedo escribir otra cosa que gracias, y justamente agradecer a todos por existir, por ser capaces de apoyarme, por callar, por hablar, por ser mis amigos aunque sea una pequeña manera de demostrarle que en verdad valió la pena conocerles. Los quiero mucho y nunca los olvidaré. A todos en general muchas gracias.



la UCI por el privilegio de poder formarme como Ingeniero.

A mi familia, sin la cual habría sido imposible llegar hasta aquí.

A Anita, mi mejor amiga, no cabrían aquí todas las palabras del mundo para agradecerte todo lo que has hecho por mí, te quiero mucho.

A “Mami-UCI” María Elena, ella sabe cuán importante ha sido en mi formación.

Al profesor Ariel Díaz Rodríguez por su ayuda decisiva e incondicional. A Erick y Rober por sus conocimientos.

A mi compañera de tesis, mi hermanita, gracias por los consejos, por darme tanto cariño y por todas las atenciones conmigo, no hubiera querido otra compañera mejor.

A mis profesores, principalmente Dailyn que nos ha enseñado a ser una familia.

A mis amigos de apartamento, especialmente Julio que ha sido mi hermano en estos 5 años, a él le debo muchos de mis logros. Al Leo (Hellboy), gracias por todos los momentos de alegría, por todas las locuras, a Yunior, el Yero, Adrián, el Yuma (Eule), Edysnay, Jesús, Agustín.

A mis compañeras de aula: Yane, Jessi, Lia, Lili, Clari, Yessenia, Lisette, Yadima, Zuleyda, Lisy, Annet. Al Flako, por dedicar parte de su tiempo libre a ayudarme con el diseño de la aplicación y por sus clases de baile. A nuestra mamá del edificio Noris, la mejor de todas las tías y la más linda, gracias por ser tan comprensiva con nosotros.

A mis brothers de la tierra: Jorge César, Ariel, Yerandy y Noslén. A mis compañeros de aula de 1er año: Roberto, Blas, Yasmany, Efrén. A mi equipo de dota, mi pikete, los mejores y más locos: Miguel (Kruznik), Fredy (Pixie), Ifraín (Ichigo), Julio (Diehard) y Melvin (Itachi), siempre podrán contar con su capitán. A Hismel, el Zami, Geiber, Hassan, Jose (Scrat), Teymor (Donka), los f9, los Gears, los Wcf, Michel (Zero), Dayán (Nazgul), Manuel (Double-Tres), Evelio (Hayabuza), Pedro (JutiaLoca), Daniel (Playmate), es imposible olvidarme de Uds....

A Icefrog y a todos los seguidores del DOTA. En general a todos aquellos que me ayudaron en esta UNIVERSIDAD.



Resumen

En la Universidad de las Ciencias Informáticas (UCI) el proceso de gestión de información es un elemento clave para lograr un mayor desarrollo y eficiencia en sus proyectos alcanzando un alto nivel en cultura organizacional en sus centros laborales y en su funcionamiento. Debido a ello la presente investigación tiene como objetivo desarrollar un Sistema de Gestión de Información de Recursos Humanos para la Dirección de Producción Audiovisual del Centro FORTES, teniendo en cuenta la importancia que tiene contribuir a perfeccionar el proceso de gestión de información en dicha organización. Para llevarlo a cabo se analizan los sistemas existentes en la actualidad y se realiza un estudio para demostrar la posibilidad de la creación de una aplicación de este tipo. Para el desarrollo de la propuesta se utilizan herramientas informáticas y una metodología de desarrollo de software como guía para su implementación. Como resultado se obtiene una aplicación donde se puede consultar y gestionar la información relacionada con los trabajadores de la Dirección de Producción Audiovisual del Centro FORTES así como controlar la asistencia y sanciones, reportar su evaluación mensual y generar diferentes modelos referentes a los trabajadores que radican en la entidad.

Índice de Tablas

Introducción.....	1
Capítulo I. Fundamentación Teórica.....	5
1.1 Introducción.....	5
1.2 Conceptos aplicados al problema de investigación.....	5
1.3 Sistemas de gestión de información.....	5
1.4 Sistemas de gestión de información para las empresas.....	6
1.5 Los Sistemas de Información a nivel internacional.....	7
1.6 Los Sistemas de Información a nivel nacional.....	8
1.7 Universidad de las Ciencias Informáticas.....	9
1.8 Herramientas informáticas y tecnologías propuestas.....	10
1.8.1 Metodologías para el desarrollo del software.....	10
1.8.2 Lenguajes de programación.....	15
1.8.3 IDE de desarrollo.....	19
1.8.4 Sistemas de Gestión de Base de Datos (SGBD).....	21
1.9 Conclusiones.....	23
Capítulo II. Características del Sistema.....	24
Fase de Exploración y Planificación.....	24
2.1 Introducción.....	24
2.2 Objeto de automatización.....	24
2.2.1 Información que se maneja.....	25
2.3 Propuesta del sistema.....	25
2.4 Personas relacionadas con el sistema.....	25
2.5 Exploración.....	26
2.5.1 Lista de Reserva del Producto (LRP).....	26
2.5.3 Requisitos no funcionales del sistema.....	28

Índice de Tablas

2.5.4 Historias de usuario.....	29
2.6 Planificación.....	35
2.6.1 Estimación de esfuerzo por cada una de las historias de usuario identificadas.....	36
2.6.2 Plan de Iteraciones.....	37
2.6.3 Plan de entrega.....	39
2.7 Conclusiones.....	40
Capítulo III. Construcción de la solución propuesta y Prueba	41
3.1 Introducción.....	41
3.2 Diseño.....	41
3.2.1 Tarjetas CRC.....	41
3.2.2 Arquitectura en tres capas.....	46
3.3 Implementación.....	48
3.3.1 Tareas de implementación para cada módulo.....	48
3.4 Pruebas.....	56
3.4.1 Introducción.....	56
3.4.2 Pruebas de aceptación.....	56
3.4.3 Pruebas de aceptación para cada historia de usuario.....	58
3.5 Conclusiones.....	64
Conclusiones generales.....	65
Recomendaciones.....	66
Referencias Bibliográficas	¡Error! Marcador no definido.
Glosario de Términos.....	¡Error! Marcador no definido.

Índice de Tablas

Tabla 1	Personas relacionadas con el sistema.	26
Tabla 2	. Funcionalidades del Sistema.	28
Tabla 3	Historia de usuario. Controlar capacitación, estudios y sanciones de los trabajadores.	30
Tabla 4	Historia de usuario. Cargar Modelos.	31
Tabla 6	Historia de usuario. Controlar Cumpleaños.	31
Tabla 7	Historia de usuario. Realizar búsquedas filtradas.	32
Tabla 8	Historia de usuario. Exportar a formatos estándares como pdf, xls, doc, html, sql y xml las planillas, modelos, actas e imprimirlas de forma directa.	32
Tabla 9	Historia de usuario. Controlar asistencia diaria mediante un scanner de barras.	33
Tabla 10	Historia de usuario. Generar acta de admisión.	33
Tabla 11	Historia de usuario. Generar boleta de admisión.	34
Tabla 12	Historia de usuario. Generar acta de advertencia.	34
Tabla 13	Historia de usuario. Generar modelo de baja.	34
Tabla 14	Historia de usuario. Generar certificado médico.	35
Tabla 15	Historia de usuario. Generar acta de amonestación	35
Tabla 16	Estimación de esfuerzo de la historias de usuario.	37
Tabla 17	Plan de iteración.	39
Tabla 18	Historias de usuario por módulos.	39
Tabla 19	Plan de entrega.	40
Tabla 20	Tarjeta CRC de Base de Datos.	42
Tabla 21	Tarjeta CRC de Modelos.	43
Tabla 22	Tarea 1 de Implementación. Insertar trabajador	48
Tabla 23	Tarea 2 de Implementación. Modificar trabajador	49
Tabla 24	Tarea 3 de Implementación. Eliminar trabajador	49
Tabla 25	Tarea 4 de Implementación. Controlar asistencia	50
Tabla 26	Tarea 5 de Implementación. Exportar a formatos estándares	50
Tabla 27	Tarea 6 de Implementación. Realizar búsquedas filtradas	51

Índice de Tablas

Tabla 28 Tarea 7 de Implementación. Cargar modelos	51
Tabla 29 Tarea 8 de Implementación. Mostrar acta de Admisión	52
Tabla 30 Tarea 9 de Implementación. Mostrar boleta de admisión.....	52
Tabla 31 Tarea 10 de implementación Generar Acta de Advertencia.	53
Tabla 32 Tarea 11 de implementación Generar modelo de baja.....	53
Tabla 33 Tarea 12 de implementación Generar evaluación.....	53
Tabla 34 Tarea 13 de implementación Generar acta de consejo de dirección.	54
Tabla 35 Tarea 14 de Implementación. Generar evaluación mensual	54
Tabla 36 Tarea 15 de Implementación. Generar modelo de asistencia	55
Tabla 37 Tarea 16 de Implementación. Generar acta de amonestación.....	55
Tabla 38 Tarea 17 de Implementación. Generar certificado médico.....	56
Tabla 39 Caso de Prueba de aceptación 1. Insertar trabajador.....	58
Tabla 40 Caso de Prueba de aceptación 2 Modificar trabajador	59
Tabla 41 Caso de Prueba de aceptación 3. Eliminar trabajador.....	60
Tabla 42 Caso de Prueba de aceptación 4. Controlar capacitación	61
Tabla 43 Caso de Prueba de aceptación 5. Controlar sanciones de los trabajadores.....	61
Tabla 44 Caso de Prueba de aceptación 6. Evaluación Docente.....	62
Tabla 45 Caso de Prueba de aceptación 7. Evaluación trabajador.....	63
Tabla 46 Caso de Prueba de aceptación 8. Cargar modelos.....	63
Tabla 47 Caso de Prueba de aceptación 9. Registrar Consejos de Dirección	¡Error! Marcador no definido.
Tabla 48 Caso de Prueba de aceptación 10. Registrar sanciones administrativas	¡Error! Marcador no definido.
Tabla 49 Caso de Prueba de aceptación 11. Controlar cumpleaños.....	¡Error! Marcador no definido.
Tabla 50 Caso de Prueba de aceptación 12. Mostrar trabajadores que estudien en la educación superior	¡Error! Marcador no definido.
Tabla 51 Caso de Prueba de aceptación 13. Mostrar trabajadores internos.....	¡Error! Marcador no definido.
Tabla 52 Caso de Prueba de aceptación 14. Conocer cantidad de trabajadores según cargo.....	¡Error! Marcador no definido.

Índice de Tablas

Tabla 53 Caso de Prueba de aceptación 15. Conocer cantidad de trabajadores según evaluación.....¡Error!

Marcador no definido.

Tabla 54 Caso de Prueba de aceptación 16. Exportar a formatos estándares¡Error! **Marcador no definido.**

Tabla 55 Caso de Prueba de Aceptación 17. Registrar hora de entrada y salida de la mañana¡Error!

Marcador no definido.

Tabla 56 Caso de Prueba de Aceptación 18. Registrar hora de entra y salida en la tarde¡Error! **Marcador no**

definido.

Tabla 57 Caso de Prueba de Aceptación 19. Generar modelo de asistencia¡Error! **Marcador no definido.**

Tabla 58 Caso de Prueba de Aceptación 20. Generar acta de admisión.....¡Error! **Marcador no definido.**

Tabla 59 Caso de Prueba de Aceptación 21. Generar boleta de admisión¡Error! **Marcador no definido.**

Tabla 60 Caso de Prueba de Aceptación 22. Generar acta de advertencia¡Error! **Marcador no definido.**

Tabla 61 Caso de Prueba de Aceptación 23. Generar modelo de baja¡Error! **Marcador no definido.**

Tabla 62 Caso de Prueba de Aceptación 24. Generar certificado médico.....¡Error! **Marcador no definido.**

Tabla 63 Caso de Prueba de Aceptación 25. Generar acta de amonestación.....¡Error! **Marcador no definido.**

Índice de Figuras

Figura 1 Logo SharpDevelop	20
Figura 2. Diagrama de Clases Persistentes.....	44
Figura 3. Modelo de Datos.....	45
Figura 4. Arquitectura en tres capas.	47

Introducción

Los cambios acelerados y el desarrollo en el ámbito de las tecnologías de la información y las comunicaciones (TICs) han ocasionado fuertes reestructuraciones en los espacios de trabajo de los diversos campos de la actividad humana, donde la producción masiva de documentación y las necesidad de acceder y gestionar la información son planes fundamentales en el desarrollo de la ciencia y la tecnología.

Cada día las empresas se enfrentan al desafío de alcanzar niveles máximos de calidad y satisfacción de sus clientes. La correcta gestión de la información y el conocimiento está estrechamente ligada a la calidad del quehacer en una organización pues favorecen y posibilitan un análisis más eficiente de los volúmenes de datos que se manejan en la misma y así descubrir patrones de comportamiento que pueden elevar la eficiencia de la entidad en la toma de decisiones organizacionales. La búsqueda de métodos para gestionar información en la actualidad ha obtenido primacía pues cada vez es mayor la cantidad de información que se precisa manejar.

Internacionalmente con todo el desarrollo tecnológico existente y sobre todo en las tecnologías de la información es muy poco probable que una empresa no utilice la información para obtener logros en sus actividades diarias. Por esta razón ha surgido la necesidad de desarrollar aplicaciones informáticas, un ejemplo de ello es el surgimiento de los sistemas de gestión de información.

Un sistema de gestión para la información tiene como objetivo lograr una mayor eficiencia en las entidades donde se emplee ya que permite que estas mejoren sus procesos internos; ejerce una influencia notable en la cultura organizacional, en los modelos laborales y en su funcionamiento. Su implantación logra ventajas competitivas y una mejor calidad de vida pues posibilita reducir los riesgos en la administración de las organizaciones, como son la toma de decisiones apresuradas, tardías o inconsistentes, que ocasionan entre otros, pérdidas de tiempo e incumplimiento del logro de los objetivos de cualquier entidad.

Introducción

La gestión de información para la Universidad de las Ciencias Informáticas (UCI) es un elemento clave para lograr un mejor desarrollo y eficiencia en sus procesos. Mientras la UCI cuenta con un alto nivel tecnológico la gestión de la información para Recursos Humanos de la Dirección de Producción Audiovisual del Centro de Tecnologías para la Formación (FORTES) realiza todos sus procesos de forma manual, por lo que resulta difícil y complejo el control de la información, además esta se almacena de forma manual provocando pérdidas y deterioro de la misma. Igualmente al consultar los datos resulta engorroso por la gran cantidad de información que se tiene y se pierde mucho tiempo para obtener la información.

Por todo lo anteriormente expuesto se define como **problema de la investigación**: ¿Cómo contribuir a la gestión de la información para Recursos Humanos de la Dirección de Producción Audiovisual del centro FORTES?

Para darle solución al problema antes mencionado se propone como **objetivo general**: desarrollar un sistema que permita gestionar todo el proceso de la información para el departamento de Recursos Humanos de la Dirección de Producción Audiovisual del Centro FORTES. Se identificaron los siguientes **objetivos específicos**:

- 1- Realizar un estudio del estado del arte.
- 2- Determinar las funcionalidades a implementar que debe cumplir el sistema.
- 3- Analizar, diseñar e implementar el sistema acorde con los requisitos planteados.
- 4- Probar la validez de la solución.

El **objeto de estudio** de la investigación se centra en los procesos de gestión de la información para Recursos Humanos. El objeto de estudio se enmarca en el **campo de acción**: los sistemas para la gestión de información de recursos humanos y se sustenta como **idea a defender**: el análisis, diseño e implementación de un sistema de gestión de información de Recursos Humanos posibilitará un rápido flujo de información relacionada con esta área de la Dirección de Producción Audiovisual del centro FORTES.

Para cumplir con los objetivos y resolver el problema de investigación se proponen las siguientes **tareas de investigación**:

- 1- Análisis de sistemas de gestión de Recursos Humanos.

- 2- Análisis de las metodologías, herramientas y métodos para el desarrollo de software.
- 3- Identificar los requisitos a implementar.
- 4- Determinar los módulos a implementar.
- 5- Desarrollar el análisis del sistema.
- 6- Desarrollar el diseño del sistema.
- 7- Implementar los módulos identificados.
- 8- Realizar pruebas al producto.

Métodos científicos utilizados en la investigación:

Métodos teóricos.

Analítico-sintético: Se utilizó en el análisis y consulta de diferentes fuentes bibliográficas acerca de las teorías existentes abordados por diferentes autores para la recopilación de información sobre los sistemas de gestión de información.

Histórico-lógico: Se utilizó este método en la investigación con el objetivo de obtener el conocimiento necesario acerca del tema tratado y su desarrollo hasta la actualidad para sistemas de gestión de información. Mediante este método se lleva a cabo un estudio sobre los antecedentes de los sistemas de gestión de información, y partiendo de la información obtenida se logra arribar a una propuesta de solución que permita asegurar la calidad del producto.

Métodos Empíricos.

Entrevista: En el presente trabajo se le realizan entrevistas a los diferentes clientes que serán beneficiados con el sistema con el objetivo de identificar sus necesidades y determinar las principales funcionalidades de la aplicación.

Este trabajo consta de varios capítulos, que reflejan las decisiones tomadas para la solución del problema, quedando estructurado de la siguiente manera:

Capítulo 1: Fundamentación Teórica. Contiene la fundamentación teórica del tema donde se exploran soluciones de sistemas de información existentes a nivel internacional, nacional y en la UCI. Además se proponen las herramientas informáticas posibles a utilizar para el desarrollo de la aplicación.

Introducción

Capítulo 2: Características del Sistema. Fases de Exploración y Planificación. En este capítulo se aborda todo lo relacionado con las características del sistema, plantea una propuesta para la solución del mismo y se definen los requisitos funcionales y no funcionales, además se modela el sistema, describiendo la manera en que se realizan las labores que serán automatizadas.

Capítulo 3: Construcción de la solución de la Propuesta y Pruebas. En este capítulo se exponen los elementos relacionados con la implementación y la realización de las pruebas de aceptación al software para evaluar el funcionamiento descrito en las historias de usuario.

Capítulo I. Fundamentación Teórica

1.1 Introducción.

En el presente capítulo se hace un estudio de los sistemas de gestión de información a nivel internacional, nacional y en la UCI. También se hace una propuesta de las herramientas informáticas y tecnologías posibles a utilizar para la solución del problema planteado, así como la justificación de la selección de las mismas.

1.2 Conceptos aplicados al problema de investigación.

Base de Datos

Una base de datos es una colección de archivos relacionados que permite el manejo de la información de una organización. Cada uno de dichos archivos puede ser visto como una colección de registros y cada registro está compuesto de una colección de campos. Cada uno de los campos de cada registro permite llevar información de algún atributo de una entidad del mundo real (1).

Sistemas de Gestión de Base de Datos (SGBD).

Los sistemas de Gestión de Base de Datos, son aplicaciones que permiten a los usuarios definir, crear y mantener la base de datos y proporciona un acceso controlado a la misma. Los SGBD son las aplicaciones que interactúan con los usuarios de los programas de aplicaciones y la base de datos. Algunos de los SGBD más conocidos son PostgreSQL, MySQL, ORACLE, DBASE, ACCESS, entre otros (2).

1.3 Sistemas de gestión de información.

La gestión de información está definida como el proceso que se encarga de gestionar la información necesaria para la toma de decisiones y un mejor funcionamiento de los procesos, productos y servicios de una organización. La correcta gestión de información conoce, incorpora y vincula todos los tipos de datos,

Capítulo I. Fundamentación Teórica

de todas las áreas de la organización y se relaciona con todos los procesos, desde la generación de datos internos, la selección y adquisición de documentos hasta la organización en su uso (3).

Tiene como principales funciones:

- Determinar necesidades internas de información, relativas a las funciones, actividades y procesos administrativos de la organización y a su administración.
- Mejorar el flujo organizacional de la información y el nivel de la comunicación.
- Manejar eficientemente los recursos organizacionales de información, mejorar las inversiones sucesivas en los mismos y optimizar su aprovechamiento.
- Entrenar a los miembros de la organización en el manejo o la utilización de los recursos informacionales.
- Contribuir a modernizar u perfeccionar las actividades organizativas y los procesos administrativos relacionados con los mismos.
- Garantizar la calidad de los productos de la organización y asegurar su disseminación efectiva.
- Determinar las necesidades de información externa de la organización y satisfacerlas.

1.4 Sistemas de gestión de información para las empresas.

Desde hace ya algunos años las organizaciones han reconocido la importancia de administrar sus recursos, por más pequeña que sean necesitan saber su pasado, la situación actual en la que se encuentran y con qué cuentan para afrontar el futuro. La información es uno de los principales recursos que poseen las empresas actualmente. Las personas que se encargan de la toma de decisiones han comenzado a comprender que la información no es sólo un subproducto de la conducción empresarial, sino que a la vez sustenta a los negocios y puede ser uno de los tantos factores críticos para la determinación del éxito o fracaso de éstos (4).

En la actualidad, para las empresas el manejo de información representa un factor de éxito. Aquellas personas encargadas de manejar organizaciones grandes y complejas dependen fuertemente de información para conocer lo que ocurre, lo que cambia y los rumbos hacia los que se dirigen los cambios. La información constituye el medio clave para manejar otros recursos como las personas, la energía, el

Capítulo I. Fundamentación Teórica

capital y las instalaciones. Por esta razón los sistemas de información en las empresas tienen una alta prioridad y son base de su eficiencia.

1.5 Los Sistemas de Información a nivel internacional.

En el mundo existen diversos sistemas que gestionan la información del personal que labora en una empresa determinada, entre estos se pueden citar los siguientes sistemas: Empleados MSD Multiusuario versión 3.2, Wikimetal RRHH versión 3.0 y @GestRRHH versión 2.0. Estos sistemas fueron diseñados para mejorar el alcance de los objetivos de los empleados de una empresa.

Sus características son:

Empleados MSD Multiusuario v3.2: Avanzado gestor de información de los empleados de una empresa que permite manejar datos personales, familiares, historial, documentos y titulaciones entre otros servicios. Este sistema está estructurado por cuatro módulos: Módulo Empleados, Módulo Información de Empleados, Módulo Resumen, Módulo Estadísticas. Es una aplicación fácil de utilizar pues presenta una interfaz sencilla y permite configurar muchas de sus características para adaptarse a las necesidades del usuario, además de permitirle un alto grado de privacidad y de aportarle herramientas de suma utilidad tales como copias de seguridad, procesador de textos, calculadora y otras (5).

Wikimetal RRHH v3.0: Sistema destinado a la gestión del personal de una empresa. Es una aplicación simple y de fácil manejo que ha sido creada con el fin de ayudar a las empresas en la difícil tarea del control de sus empleados. Entre otras funciones, brinda la posibilidad de gestionar las vacaciones de los empleados de forma mensual, bimensual o trimestral. También se puede efectuar un control por departamentos e incluso visualizar inmediatamente los retrasos, ausencias o sanciones de los empleados (6).

@GestRRHH v2.0: Sistema diseñado para la gestión de recursos humanos compuesto por los siguientes módulos: Agenda, Plantilla, Actividades y Nóminas. Esta herramienta permite la realización de un gran número de informes asociados con cada uno de sus módulos. Estos últimos poseen un color individual para mejorar la información y permite la copia de seguridad automática (7).

Capítulo I. Fundamentación Teórica

Los sistemas analizados ofrecen una visión de cómo es gestionada la información de los recursos humanos de forma general y ayudan al entendimiento de los procesos que estos implementan, pero no se ajustan estrictamente a las necesidades del sistema a desarrollar pues no cuentan con funcionalidades para controlar la asistencia diaria mediante un lector de código de barras, reportar la evaluación mensual así como generar diferentes modelos relacionados con los trabajadores como acta de admisión, modelo de baja, certificado médico entre otros. Además no se puede adquirir porque el costo de su licencia es muy alto.

1.6 Los Sistemas de Información a nivel nacional.

En Cuba existen Sistemas de Registro y Control de Personal para mantener organizado todo su capital humano entre los que se pueden citar los siguientes: Software RH-CITMA y Registro Personal de la Salud. A continuación se explica algunas características de ambos.

Software RH-CITMA v1.0: Pretende obtener las ventajas de la identificación de los procesos de la empresa para la gestión de algunas actividades de recursos humanos. Este sistema esta compuesto por diferentes módulos relacionados entre sí entre los que se encuentran: Diseño de empresa, Procesos, Selección del Personal, Formación, Inventario de persona entre otros. RH-CITMA fue implantado en el Centro de Información y Gestión Tecnológica de Cienfuegos. Empresa perteneciente al CITMA (Ciencia Tecnología y Medio Ambiente) (8).

Registro Personal de la Salud: Basado en software libre y en las nuevas tecnologías de Internet como XML, PHP y Servicios Web que garantizan una explotación integrada y compatible. Utiliza como soporte de esta aplicación la infraestructura de INFOMED. Cumple con la arquitectura definida por el Grupo de Arquitectura MINSAP–Ministerio de Informática y las Comunicaciones (MIC), desarrollando productos informáticos sobre una arquitectura basada en componentes y orientada a servicios, haciendo uso de la plataforma de servicios (plaser) y los Servicios Web (9).

Este sistema tiene las siguientes características:

- Almacena información en una base de datos única, sólida y confiable.

Capítulo I. Fundamentación Teórica

- Permite gestionar al personal de salud pendiente de ubicar en una unidad de salud, abarcando los movimientos de traslado, jubilación y baja, así como los movimientos de recién graduados.
- Realiza búsquedas de profesionales por características, que pueden ser: nombre, tipo de profesional, nivel técnico entre otros.

Estas aplicaciones no se utilizan porque no cumplen en su totalidad con las características del sistema a desarrollar ya que no presentan funcionalidades para controlar la asistencia, controlar la capacitación, estudios y sanciones de los trabajadores entre otros.

1.7 Universidad de las Ciencias Informáticas.

En la Universidad existen algunos sistemas que gestionan información de personas, dentro de los que se pueden citar:

Sistema de Registro Integral (ASSETS): Sistema que registra todo el personal que es plantilla en la UCI. Este sistema es del tipo de Planificación de Recursos Empresariales (ERP), de factura foránea. Se introdujo en Cuba desde el año 1997. Controla plenamente los recursos laborales, dígame: empleados, estructura organizativa de la entidad y plantilla. Es válido aclarar que este sistema no se desarrolló en la UCI, sino que se modificó teniendo en cuenta el personal que se gestiona en la universidad proceso propio de la misma. La aplicación consta de diferentes módulos, así como finanzas, control de inventario, activos fijos, recursos humanos y nómina, entre otros.

El Sistema de Registro de Personal: Está enmarcado en el control del personal externo de la UCI. Las funcionalidades que brinda el sistema son:

- Autenticar Usuario.
- Gestionar Persona.
- Gestionar Áreas Tercerizadas.
- Gestionar Atributos por categorías.
- Gestionar Categorías de personas.
- Gestionar Funcionalidades.
- Gestionar Roles.

Capítulo I. Fundamentación Teórica

- Mostrar Roles.
- Verificar Personas Registradas.

Estos sistemas no cumplen exactamente con las necesidades que presenta el departamento de recursos humanos de la Dirección de Producción Audiovisual del Centro FORTES porque no presentan un control exactamente conforme con las características requeridas como reportar asistencia mediante un lector de código de barras y gestionar diferentes modelos relacionados con los trabajadores, así como el modelo de baja, acta de admisión, acta de advertencia, certificado médico entre otros.

1.8 Herramientas informáticas y tecnologías propuestas.

El empleo de herramientas informáticas aumenta la productividad y competitividad de los proyectos de desarrollo de software. En este sentido, la incorporación de nuevas tecnologías a los procesos de producción o prestación constituye un desafío. La selección de las mismas requiere de un estudio detallado y crítico, teniendo en cuenta las necesidades del cliente y los recursos con los que se dispone durante el proceso de desarrollo de software.

1.8.1 Metodologías para el desarrollo del software.

El proceso de desarrollo de software es riesgoso y difícil de controlar, pero si no tiene una metodología de por medio aumenta la probabilidad de que pueda traer como resultado clientes descontentos con el producto y desarrolladores aún más insatisfechos. Existen numerosas propuestas de metodologías que inciden en distintas dimensiones del proceso de desarrollo. Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente.

Metodologías tradicionales.

Las metodologías tradicionales centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto, todo esto definido, en la fase inicial del desarrollo del proyecto.

Capítulo I. Fundamentación Teórica

Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado comienza el ciclo de desarrollo del producto de software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos pueden variar (10).

Las metodologías tradicionales se caracterizan por:

- Estar basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
- Cierta resistencia a los cambios.
- Proceso mucho más controlado, con numerosas políticas y normas.
- Existe un contrato prefijado.
- El cliente interactúa con el equipo de desarrollo mediante reuniones.
- Grupos grandes y posiblemente distribuidos.
- La arquitectura del software es esencial y se expresa mediante modelos. (11)

Metodologías ágiles.

Las metodologías ágiles, dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque muestra su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software y generan un amplio debate entre sus seguidores y quienes no las ven como alternativa para las metodologías tradicionales (12).

Las metodologías ágiles se caracterizan por:

- Heurísticas provenientes de prácticas de producción de código.
- No existe contratos tradicionales o al menos son bastante flexibles.
- Los clientes forman parte del equipo de desarrollo.

Capítulo I. Fundamentación Teórica

- Grupos pequeños (menos de 10 integrantes) y trabajando en el mismo sitio.
 - Hacen menos énfasis en la arquitectura de software que las metodologías tradicionales.
- (13)

Una vez realizado el estudio sobre las posibles metodologías de desarrollo de software se seleccionó por parte del equipo de trabajo una metodología ágil pues se enfocan más en desarrollar software que gocen de éxito más que en lograr una buena documentación. La colaboración con el cliente más que la negociación de un contrato pues se propone que exista una interacción constante entre el cliente y el equipo de desarrollo, esta colaboración será la que marque la marcha del proyecto y asegure su éxito. Responder a los cambios más que seguir estrictamente un plan ya que la habilidad de responder a los cambios que pueden surgir a lo largo del proyecto (cambios en los requisitos, tecnologías entre otros) determina también el fracaso o el éxito del mismo. Dentro de las metodologías ágiles se encuentra:

Scrum.

Scrum es una metodología ágil para el desarrollo de productos, que se centra en priorizar el trabajo en función del valor que tenga para el negocio. Aunque surgió como modelo para el desarrollo de productos tecnológicos, se emplea en entornos que trabajan con requisitos inestables y que requieren agilidad y flexibilidad, situaciones frecuentes en el desarrollo de software. El desarrollo de aplicaciones y la gestión de proyectos tecnológicos, además de los conceptos básicos de calidad, coste y diferenciación, exige también rapidez y flexibilidad, por lo que la aplicación de esta metodología mejora el control, la gestión y la rentabilidad de los proyectos (14).

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Scrum permite la creación de equipos organizados impulsando la localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto (15).

Programación Extrema (eXtreme Programming, XP).

XP es una de las metodologías ágiles de desarrollo de software para proyectos de corto plazo y equipos pequeños. La metodología formulada por Kent Beck consiste en una programación rápida o extrema, cuya

Capítulo I. Fundamentación Teórica

particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Las características fundamentales de la metodología XP son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir una nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se introduzca ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: la metodología XP apuesta que es más sencillo hacer algo simple y aunque requiera de trabajo extra para cambiarlo, que realizar algo complicado y quizás nunca utilizarlo. La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores (16) .

Las ventajas de esta metodología según Solís y otros autores son (17):

- Apropiaada para entornos volátiles.
- Estar preparados para el cambio, significa reducir su coste.
- Planificación más transparente para los clientes, éstos conocen las fechas de entrega de funcionalidades vitales para su negocio.
- Permite definir en cada iteración cuáles son los objetivos de la siguiente.

Capítulo I. Fundamentación Teórica

- Posibilita tener realimentación de los usuarios.
- La presión está a lo largo de todo el proyecto y no en una entrega final.

Se selecciona la metodología ágil XP ya que se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Además los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. XP es 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. No se selecciona Scrum porque no genera toda la evidencia o documentación de otras metodologías creando así resistencia a su uso por diferentes personas, además permite fallar si es necesario cuestión que el equipo de desarrollo no considera permitirse. También esta metodología exige solo miembros de equipo experimentados.

Herramientas CASE.

Visual Paradigm 3.4.

Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Computación). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta de software libre de utilidad para el analista. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos (18).

Se caracteriza por:

1. Disponibilidad en múltiples plataformas (Windows, Linux).

Capítulo I. Fundamentación Teórica

2. Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
3. Distribución mediante licencia gratuita y comercial.
4. Utilización de varios idiomas.
5. Facilidad de instalación y actualización.

Rational Rose 7.0.

Es una herramienta CASE desarrollada por Rational Corporation basada en el Lenguaje Unificado de Modelado (UML), que permite crear los diagramas que se van generando durante el proceso de ingeniería en el desarrollo del software. Rational permite completar una gran parte de las disciplinas (flujos fundamentales) de RUP tales como:

1. Captura de requisitos.
2. Análisis y diseño.
3. Implementación.
4. Control de cambios y Gestión de configuración.

Se selecciona la herramienta Visual Paradigm porque contiene un entorno de trabajo organizado, además la UCI tiene la licencia que permite el uso de la misma. No se utiliza la herramienta Rational Rose ya que es propietaria.

1.8.2 Lenguajes de programación.

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina. Además permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo determinadas circunstancias. Una característica relevante de

Capítulo I. Fundamentación Teórica

los lenguajes de programación es precisamente que más de un programador pueda tener un conjunto común de instrucciones que puedan ser comprendidas entre ellos para realizar la construcción del programa de forma colaborativa (19).

C++.

Lenguaje exigente orientado a objeto de propósito general basado en el lenguaje C, al que se han añadido nuevos tipos de datos, clases, plantillas, sistema de espacios de nombres, funciones inline, sobrecarga de operadores, referencias, operadores para manejo de memoria persistentes y algunas utilidades adicionales de bibliotecas (en realidad la biblioteca Estándar C es un subconjunto de la bibliotecas C++). C++ es un lenguaje de programación extremadamente largo y complejo (20).

Aunque este lenguaje posee características que lo hacen eficiente como la posibilidad de redefinir los operadores (sobrecarga de operadores), y de poder crear nuevos tipos que se comporten como tipos fundamentales, presenta algunas desventajas en cuanto a facilidad de uso como la utilización de librerías dinámicas muy complejas.

Java.

Java fue diseñado como un lenguaje orientado a objetos desde su surgimiento. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red. Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas. Sus características de memoria liberan a los programadores de errores (la aritmética de punteros), ya que se ha prescindido por completo de los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria (21).

Java no es un lenguaje totalmente orientado a objetos. Por motivos de eficiencia, por ejemplo, no todos los valores son objetos. El código Java puede ser a veces redundante en comparación con otros lenguajes. Esto es en parte debido a las frecuentes declaraciones de tipos y conversiones de tipo manual

Capítulo I. Fundamentación Teórica

(casting). También se debe a que no se dispone de operadores sobrecargados, y a una sintaxis relativamente simple. Sin embargo, J2SE 5.0 introduce elementos para tratar de reducir la redundancia, como una nueva construcción para los bucles "foreach". A diferencia de C++, Java no dispone de operadores de sobrecarga definidos por el usuario. Java presenta desventajas, pues al poseer clases de librerías el manejo del lenguaje se hace más difícil y consume mucho tiempo. También en ocasiones puede ocurrir un desperdicio de memoria, lo que permite que la ejecución del código sea lenta.

Visual Basic.

Visual Basic constituye un Entorno de Desarrollo Integrado (*Integrated Development Environment*, IDE) que ha sido empaquetado como un Programa de aplicación; es decir, consiste en un editor de código (programa donde se escribe el código fuente), un depurador (programa que corrige errores en el código fuente para que pueda ser bien compilado), un compilador (programa que traduce el código fuente a lenguaje de máquina), y un constructor de interfaz gráfica o GUI (es una forma de programar en la que no es necesario escribir el código para la parte gráfica del programa, sino que se puede hacer de forma visual). Con Visual Basic se pueden desarrollar aplicaciones para Windows más rápidamente. Los errores de Programación no se generan tan frecuentemente y, si lo hacen, son sencillos de depurar. Además incluye dos conceptos importantes:

- Un método visual de creación de aplicaciones, incluyendo formularios (Ventanas), controles y, componentes del formulario.
- La habilidad de asociar código directamente a cada evento de cada elemento del diseño visual. Es posible escribir aplicaciones sin usar componentes visuales, es decir escribir aplicaciones de consola (22).

El lenguaje de programación Visual Basic proporciona todas las herramientas necesarias para el desarrollo rápido de aplicaciones. Es posible definir programación en Visual Basic como el método que se utiliza para desarrollar la interfaz gráfica de usuario. La programación en este lenguaje permite construir de forma rápida aplicaciones de Windows, pero no se saca provecho de las mejores funcionalidades de la programación orientada a objetos, y no se adecua para desarrollar aplicaciones de gran envergadura.

Capítulo I. Fundamentación Teórica

C Sharp.

C# es un lenguaje orientado a objetos, moderno y seguro. La sintaxis y estructuración de C# es muy similar a la de C++, ya que la intención de Microsoft con C# es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son equiparables a los de Visual Basic (23).

Características

C Sharp presenta características como: sencillez, ya que elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo: El código escrito en C# es auto contenido, lo que significa que no necesita de ficheros adicionales de la propia fuente tales como ficheros de cabecera o ficheros IDL. El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile, lo que facilita la portabilidad del código. No se incluyen elementos poco útiles de C++ como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) para acceder a miembros de espacios de nombres (::). Otra característica que presenta es la modernidad, pues incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose, son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico decimal, la inclusión de una instrucción foreach que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, la inclusión de un tipo básico String para representar cadenas o la distinción de un tipo bool específico para representar valores lógicos. Además C Sharp es orientado a objetos diferenciando este enfoque orientado a objetos respecto al de otros lenguajes como C++ pues C# es más puro en tanto que no admite ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código. También se caracteriza por la seguridad en el manejo de datos ya que incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, evitando que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto.

Capítulo I. Fundamentación Teórica

Ventajas:

Compila a código intermedio independiente del lenguaje en que haya sido escrita la aplicación e independiente de la máquina donde vaya a ejecutarse. Se ahorra tiempo en la programación ya que presenta una librería de clases muy completa y bien diseñada. Además presenta un concepto formalizado de los métodos Get y Set, con lo que se consigue código mucho más legible (24).

¿Por qué una aplicación de escritorio?

Una aplicación de escritorio es aquella que se instala en el ordenador de un usuario, luego es ejecutada directamente por el sistema operativo y cuyo rendimiento depende de diversas configuraciones de hardware como memoria RAM, disco duro, memoria de video, etc.

Se decide desarrollar una aplicación de escritorio ya que sus características se adecúan con las necesidades del cliente pues la ejecución del sistema no requiere comunicación con el exterior, o sea conexión a la red, sino que se realiza de forma local. Esto repercute en mayor velocidad de procesamiento, y por tanto estabilidad ya que su rendimiento es mejor y el tiempo de respuesta es rápido. Su acceso se limita al ordenador donde está instalada, restricción que no repercute en el desarrollo de la aplicación, ya que existe un único usuario destinado al trabajo con el sistema.

¿Por qué C Sharp?

Se selecciona C Sharp como lenguaje de programación teniendo en cuenta que sus características se alinean a las necesidades del cliente de una aplicación que presente gran velocidad en la manipulación de los datos así como una interfaz que sea lo más rápida posible y que permita una fácil navegación, por el usuario que interactúa con ella. Además el equipo de desarrollo cuenta con suficiente experiencia en la plataforma .NET y en este lenguaje.

1.8.3 IDE de desarrollo

Un IDE es un conjunto de herramientas para el programador que incluyen un buen editor de código, administrador de proyectos y archivos, enlace transparente a compiladores e integración con sistemas controladores de versiones o repositorios (25).

Capítulo I. Fundamentación Teórica

Microsoft Visual Studio 2008.

Es un IDE para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros. Con Visual Studio Tools para Office (VSTO) integrado con Visual Studio 2008 es posible desarrollar rápidamente aplicaciones de alta calidad basadas en la interfaz de usuario (UI) de Office que personalicen la experiencia del usuario y mejoren su productividad en el uso de Word, Excel, PowerPoint, Outlook, Visio, InfoPath y Project. Aunque este entorno de desarrollo se adecúa a las características que requiere el sistema que se desea implementar el costo de su licencia es muy alto por lo que imposibilita su adquisición.

SharpDevelop

Herramienta gratuita que actúa como un entorno de desarrollo de programación basándose en lenguajes como: C#, Visual Basic .NET y Boo. Es usado típicamente por aquellos programadores de los citados lenguajes, que no desean o no pueden usar el entorno de desarrollo de Microsoft, el Microsoft Visual Studio.

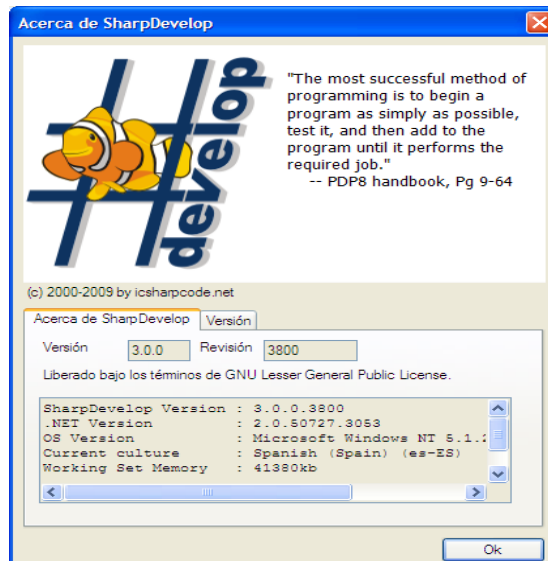


Figura 1 Logo SharpDevelop

Capítulo I. Fundamentación Teórica

Una característica muy relevante de SharpDevelop es el coloreado de sintaxis del lenguaje C. Lo cual significa algo muy importante para poder diferenciar fragmentos del código. Sharp Develop integra una gran diversidad de plantillas de códigos de programas ya predeterminadamente creados para añadir o crear ficheros, proyectos o compiladores. Esto sirve mucho para aquellos que ya conozcan programación avanzada pues el trabajo se les hace más fácil y automatizado. También se incluye un buscador de referencias dentro del mismo código, como un asistente de reemplazamiento de fragmentos del código. El programa tiene completa disponibilidad de estar en español. Y puede ser extensible con plugins que están disponibles en su misma página oficial como también algunas herramientas adicionales lo cual hace que el programa sea más práctico y completo. Por las características, anteriormente expuestas, que presenta este entorno se decide utilizarlo como IDE de desarrollo.

1.8.4 Sistemas de Gestión de Base de Datos (SGBD).

Los SGBD, son aplicaciones que permiten a los usuarios definir, crear y mantener una base de datos y proporciona el acceso controlado a la misma, además permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Algunos de los SGBD más conocidos son: PostgreSQL, MySQL, ORACLE, DBASE, ACCESS, entre otros (2).

El mercado del software mundial está dominado por compañías multimillonarias donde insertarse con productos de calidad es difícil. Sin embargo en este mundo el Software Libre se ha ganado un espacio importante y cada día son más los desarrolladores que usan estas herramientas para hacer sus aplicaciones. Esto fue tomado en cuenta para escoger un servidor de bases de datos para la aplicación, pues la mayoría de las herramientas élites en el mundo son propietarias como: SQL Server, Oracle y Microsoft Access, lo que implica un alto costo de su licencia. Dentro de los gestores libres se destacan MySQL y PostgreSQL.

PostgreSQL.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional. Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. Incluye características de la orientación a objetos, como puede ser la herencia, tipos

Capítulo I. Fundamentación Teórica

de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos (26). Por otra parte no cuenta con un conjunto de herramientas que permitan una fácil gestión de los usuarios y de las bases de datos que contenga el sistema. Además la velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede parecer deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes.

MySQL 5.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (viadbExpress), Java (con una implementación nativa del driver de Java), PHP, Python; cada uno de estos utiliza una API específica. También existe una interfaz ODBC (estándar de acceso a base de datos), llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL (21).

MySQL es muy utilizado tanto en aplicaciones web como de escritorio, en plataformas (Linux/Windows-Apache-MySQL-PHP/Python), permite manejar registros de longitud fija o variable. Además posee muchas ventajas funcionales, entre las que se pueden mencionar las siguientes (27):

- ✓ Acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.
- ✓ Seguridad, en forma de permisos y privilegios, determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas. Esto permite compartir datos sin que peligre la integridad de la base de datos o protegiendo determinados contenidos.
- ✓ Potencia: SQL es un lenguaje muy potente para consulta de bases de datos.

Capítulo I. Fundamentación Teórica

- ✓ Portabilidad: SQL es también un lenguaje estandarizado, de modo que las consultas hechas usando SQL son fácilmente portables a otros sistemas y plataformas. Esto, unido al uso de C/C++ proporciona una portabilidad enorme.

Después de realizar un estudio sobre los dos gestores de base de datos más reconocidos se puede constatar que MySQL es el más idóneo para usar en el desarrollo de la aplicación. A continuación una serie de ventajas de MySQL sobre PostgreSQL, teniendo en cuenta las características de la aplicación que se desea desarrollar:

- Mayor velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrece mayor rendimiento.
- Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Presenta grandes utilidades de administración debido a su facilidad de configuración e instalación.
- Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.
- No hay límites en el tamaño de los registros.

1.9 Conclusiones.

En este capítulo se ofreció información sobre varias aplicaciones del mundo con características similares al sistema que se propone para resolver el problema de la investigación. Se describieron además de una manera simple las herramientas y tecnologías propuestas para el desarrollo de la aplicación donde se muestran las principales características y ventajas que condujeron a la selección de las mismas.

Capítulo II. Características del Sistema

Fase de Exploración y Planificación

2.1 Introducción.

En este capítulo se describen los procesos que se deben automatizar en el departamento de Recursos Humanos de la Dirección de Producción Audiovisual del Centro FORTES como parte de la propuesta de solución al problema planteado. Se presentan las fases de Planificación y Exploración de la metodología XP teniendo como resultado las historias de usuario creadas por el cliente y toda la planificación para darle cumplimiento a estas funcionalidades.

2.2 Objeto de automatización.

El Departamento de Recursos Humanos de la Dirección de Producción Audiovisual del Centro FORTES maneja información de vital importancia para su funcionamiento. Recopilar estos datos para obtener un resultado se vuelve difícil, de ahí la necesidad de automatizar los procesos que se realizan en el departamento como:

- Gestionar trabajadores.
- Controlar capacitación, estudios y sanciones de los trabajadores.
- Reportar evaluación mensual.
 - Evaluación Docente.
 - Evaluación del Trabajador.
- Cargar Modelos.
- Registrar Consejos de Dirección.
- Controlar Cumpleaños.
- Realizar búsquedas filtradas.
 - Conocer trabajadores que estudian en educación superior.
 - Conocer los trabajadores internos.
 - Conocer cantidad de trabajadores según cargo.
 - Conocer cantidad de trabajadores según evaluación.
- Exportar a formatos estándares como pdf, xls, doc, html, sql y xml las planillas, modelos, actas e imprimirlas de forma directa.

Capítulo II. Características del Sistema

- Controlar asistencia diaria mediante un scanner de barras.
 - Registrar hora de entrada y salida de la mañana de un trabajador.
 - Registrar hora de entrada y salida de la tarde de un trabajador.
 - Generar modelo de asistencia.
- Generar acta de admisión.
- Generar boleta de admisión.
- Generar acta de advertencia.
- Generar modelo de baja.
- Generar certificado médico.
- Generar acta de amonestación.

2.2.1 Información que se maneja.

La información que se encuentra relacionada con los trabajadores de la Dirección de Recursos Humanos de la Dirección de Producción Audiovisual del Centro FORTES.

2.3 Propuesta del sistema.

Se propone el desarrollo de un Sistema de Gestión de Información para Recursos Humanos de la Dirección de Producción Audiovisual del Centro FORTES teniendo en cuenta que se realizan todos sus procesos de forma manual. Con este sistema se facilitará el trabajo con la información referente a los trabajadores de esta área y permitirá un mejor control de la misma.

2.4 Personas relacionadas con el sistema.

Las personas relacionadas con el sistema son todas aquellas que obtienen un resultado del valor de uno o varios procesos que se ejecutan en el propio sistema. De igual forma son las que se encuentran involucradas en el sistema pero no obtienen ningún valor a pesar de participar en ellos. A continuación se presentan las personas que estarán involucradas con la aplicación.

Capítulo II. Características del Sistema

Personas relacionadas con el sistema	Justificación
Asistente	La asistente es quien hace la función de administrador, es la encargada de crear las cuentas de usuario y facultada para realizar cualquier operación dentro del sistema.

Tabla 1 *Personas relacionadas con el sistema.*

2.5 Exploración.

En esta fase, los clientes plantean a grandes rasgos los procesos que se quieren automatizar a través de las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del conocimiento y familiaridad que tengan los programadores con la tecnología.

2.5.1 Lista de Reserva del Producto (LRP).

La Lista de Reserva del Producto es un documento de alto nivel para el proyecto que contiene amplias descripciones de los requisitos funcionales ordenados según la prioridad de implementación, ubicada en Muy Alta, Alta, Media y Baja, además de mencionar los requisitos no funcionales del sistema. Cuando un proyecto comienza es muy difícil tener claro todos los requisitos, el equipo de desarrollo se enfoca en las funcionalidades que por el momento suelen ser más importantes para el sistema. En el transcurso del tiempo esta lista puede crecer y modificarse debido a las constantes reuniones e intercambios entre el cliente y los desarrolladores donde adquieren mayor información sobre el producto. Al terminar de llenar la lista el objetivo es obtener un producto competitivo, lo más correcto posible y que satisfaga las necesidades del cliente. El uso de esta lista es de gran importancia pues pone en marcha una de las principales características de la metodología XP: ser recomendada para proyectos con requisitos

Capítulo II. Características del Sistema

imprecisos y muy variantes. Los programadores que utilizan esta metodología consideran que los cambios frecuentes de requisitos durante el desarrollo del software son un aspecto natural, inevitable y que poder adaptarse a estos cambios en cualquier parte del ciclo de vida del proyecto es una aproximación mejor y más realista.

2.5.2 Funcionalidades del sistema.

Funcionalidades del sistema	Prioridad
Gestionar trabajador. <ul style="list-style-type: none">• Insertar trabajador.• Modificar trabajador.• Eliminar trabajador.• Mostrar datos de un trabajador.	Muy Alta
Controlar capacitación, estudios y sanciones de los trabajadores.	Media
Reportar evaluación mensual. <ul style="list-style-type: none">• Evaluación Docente.• Evaluación del trabajador.	Alta
Cargar modelos.	Alta
Controlar cumpleaños	Baja
Realizar búsquedas filtradas. <ul style="list-style-type: none">• Conocer trabajadores que estudian en educación superior.• Conocer los trabajadores internos.• Conocer cantidad de trabajadores según cargo.• Conocer cantidad de trabajadores según evaluación.	Alta
Exportar a formatos estándares como pdf, xls, doc, html, sql y xml las planillas, modelos, actas e imprimirlas de forma directa.	Alta

Capítulo II. Características del Sistema

Controlar asistencia diaria mediante un lector de código de barras de barras. <ul style="list-style-type: none">• Registrar hora de entrada y salida de la mañana de un trabajador.• Registrar hora de entrada y salida de la tarde de un trabajador.• Generar modelo de asistencia.	Alta
Generar acta de admisión.	Media
Generar boleta de admisión.	Media
Generar acta de advertencia.	Baja
Generar modelo de baja.	Media
Generar certificado médico.	Baja
Generar acta de amonestación.	Baja

Tabla 2 . Funcionalidades del Sistema.

2.5.3 Requisitos no funcionales del sistema.

1. *Requisitos de Software.*

- La computadora personal (PC) de la asistente debe tener instalado el Framework 4.0 de .NET, el gestor de base de datos MySQL v5.0 y el sistema operativo Windows Vista/XP/2000/NT.

2. *Requisitos de Hardware.*

- La PC debe tener como mínimo 256 Mb de RAM y, 5 Gb libres en el disco duro debido a que la documentación que se va a guardar en diferentes formatos irá creciendo con el tiempo.

3. *Restricciones en el diseño y la implementación.*

- Implementación del sistema con el lenguaje de programación C#.
- Gestor de base de datos MySQL.

Capítulo II. Características del Sistema

4. Requisitos de apariencia de interfaz externa.

- Interfaz externa amigable y de navegación sencilla para el usuario.

5. Requisitos de seguridad.

- Solo podrá acceder a la aplicación la asistente del departamento de Recursos Humanos.

6. Requisitos de usabilidad.

- El sistema puede ser usado por personas que tengan conocimientos básicos sobre el manejo de un ordenador.

2.5.4 Historias de usuario

En esta etapa los principales artefactos que se obtienen son las historias de usuario, XP las utiliza como una técnica para reemplazar el documento de requisitos. Las historias de usuario son tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer, ya sean requisitos funcionales o no funcionales. Para su construcción se emplea terminología del cliente, sin lenguaje técnico. El contenido de las mismas debe ser sencillo, concreto y debe proporcionar solo el detalle suficiente como para poder hacer razonable la estimación de cuánto tiempo requiere la implementación de la misma. Se realiza una por cada característica principal del sistema. Difieren de los casos de uso porque son escritas por los clientes, no por los programadores. Las historias de usuario son la base de las pruebas de aceptación pues mediante estas se verifica si han sido implementadas correctamente y si cumplen las expectativas del cliente. Existen diferencias entre las historias de usuario y la habitual especificación de requisitos, divergiendo principalmente en el nivel de detalle.

A continuación se muestran las historias de usuario identificadas por el cliente:

Historia de usuario	
Número: 1	Nombre: Gestionar trabajador
Usuario: Asistente	

Capítulo II. Características del Sistema

Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo
Iteración asignada: 1	
Descripción: El Administrador tendrá la posibilidad de gestionar los trabajadores que sean necesarios	
Observaciones: Gestionar trabajador incluye crear, modificar y eliminar trabajador	

Tabla 3 Historia de usuario. Gestionar evaluación.

Historia de usuario	
Número: 2	Nombre: Controlar capacitación, estudios y sanciones de los trabajadores
Usuario: Asistente	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Bajo
Iteración asignada: 2	
Descripción: Brindará la posibilidad de registrar la capacitación, estudios y sanciones de los trabajadores	
Observaciones:	

Tabla 3 Historia de usuario. Controlar capacitación, estudios y sanciones de los trabajadores.

Historia de usuario	
Número: 3	Nombre: Cargar modelos
Usuario: Asistente	

Capítulo II. Características del Sistema

Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Iteración asignada: 1	
Descripción: El sistema permitirá a la asistente cargar los modelos de Evaluación de Desempeño de los trabajadores de la Dirección de Producción Audiovisual, Planificación de las Vacaciones Dirección de Producción Audiovisual y Pre-nómina 2011 Actualizada	
Observaciones:	

Tabla 4 Historia de usuario. Cargar modelos.

Historia de usuario	
Número: 4	Nombre: Controlar cumpleaños
Usuario: Asistente	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Bajo
Iteración asignada: 3	
Descripción: El sistema permitirá conocer la fecha de los cumpleaños de cada trabajador	
Observaciones:	

Tabla 5 Historia de usuario. Controlar cumpleaños.

Historia de usuario	
Número: 5	Nombre: Realizar búsquedas filtradas
Usuario: Asistente	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo

Capítulo II. Características del Sistema

Iteración asignada: 1
Descripción: Permitirá realizar búsquedas de acuerdo a la información que necesite
Observaciones:

Tabla 6 Historia de usuario. Realizar búsquedas filtradas.

Historia de usuario	
Número: 6	Nombre: Exportar a formatos estándares como pdf, xls, doc, html, sql y xml las planillas, modelos, actas e imprimirlas de forma directa
Usuario: Asistente	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo
Iteración asignada: 1	
Descripción: El sistema dará la posibilidad de exportar a diferentes formatos estándares	
Observaciones:	

Tabla 7 Historia de usuario. Exportar a formatos estándares como pdf, xls, doc, html, sql y xml las planillas, modelos, actas e imprimirlas de forma directa.

Historia de usuario	
Número: 7	Nombre: Controlar asistencia diaria mediante un lector de código de barras
Usuario: Asistente	

Capítulo II. Características del Sistema

Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo
Iteración asignada: 1	
Descripción: Permitirá controlar la asistencia diaria de los trabajadores mediante un scanner de barras	
Observaciones:	

Tabla 8 Historia de usuario. Controlar asistencia diaria mediante un lector de código de barras de barras.

Historia de usuario	
Número: 8	Nombre: Generar acta de admisión
Usuario: Asistente	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Bajo
Iteración asignada: 2	
Descripción: El sistema deberá dar la posibilidad de generar un acta de admisión	
Observaciones:	

Tabla 9 Historia de usuario. Generar acta de admisión.

Historia de usuario	
Número: 9	Nombre: Generar boleta de admisión
Usuario: Asistente	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alto
Iteración asignada: 2	
Descripción: Brindará la posibilidad de generar una boleta de admisión	

Capítulo II. Características del Sistema

Observaciones:

Tabla 10 Historia de usuario. Generar boleta de admisión.

<i>Historia de usuario</i>	
Número: 10	Nombre: Generar acta de advertencia
Usuario: Asistente	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Bajo
Iteración asignada: 3	
Descripción: El sistema permitirá generar un acta de advertencia	
Observaciones:	

Tabla 11 Historia de usuario. Generar acta de advertencia.

<i>Historia de usuario</i>	
Número: 11	Nombre: Generar modelo de baja
Usuario: La asistente	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Bajo
Iteración asignada: 2	
Descripción: El sistema brindará la posibilidad de generar un modelo de baja	
Observaciones:	

Tabla 12 Historia de usuario. Generar modelo de baja.

<i>Historia de usuario</i>

Capítulo II. Características del Sistema

Número: 12	Nombre: Generar certificado médico
Usuario: La asistente	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Alto
Iteración asignada: 3	
Descripción: El sistema brindará la posibilidad de generar un certificado médico	
Observaciones:	

Tabla 13 Historia de usuario. Generar certificado médico.

Historia de usuario	
Número: 13	Nombre: Generar acta de amonestación
Usuario: La asistente	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Alto
Iteración asignada: 3	
Descripción: El sistema brindará la posibilidad de generar un certificado médico	
Observaciones:	

Tabla 14 Historia de usuario. Generar acta de amonestación

2.6 Planificación.

En la fase de planificación se priorizan las historias de usuario, siendo el cliente quien determina esta prioridad. Por otra parte los programadores estiman cuánto esfuerzo requiere cada historia, a partir de ahí se define el cronograma y en conjunto con el cliente se toman los acuerdos de la primera entrega. La planificación se puede realizar basándose en el tiempo o el alcance y sólo toma uno o dos días aproximadamente. Una entrega debería obtenerse en no más de tres meses. La fase de planificación

Capítulo II. Características del Sistema

generalmente dura pocos días. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto.

2.6.1 Estimación de esfuerzo por cada una de las historias de usuario identificadas.

Para el desarrollo de la aplicación se realizó una estimación de esfuerzo por cada una de las historias de usuario identificadas por el cliente. Las estimaciones de esfuerzo asociado a la implementación se determinan utilizando como medida la cantidad de puntos. Se le dan valores de 1 a 3 puntos donde un punto equivale a una semana de programación. El resultado obtenido queda plasmado en la tabla que se muestra a continuación.

<i>Historias de usuario</i>	<i>Puntos estimados</i>
Gestionar trabajador.	1
Controlar capacitación, estudios y sanciones de los trabajadores.	1
Reportar evaluación mensual.	1
Cargar modelos.	1
Controlar cumpleaños	1
Realizar búsquedas filtradas.	2
Exportar a formatos estándares como pdf, xls, doc, html, sql y xml las planillas, modelos, actas e imprimirlas de forma directa.	2
Controlar asistencia diaria mediante un lector de código de barras.	3

Capítulo II. Características del Sistema

Generar acta de admisión.	1
Generar boleta de admisión.	1
Generar acta de advertencia.	1
Generar modelo de baja.	1
Generar certificado médico.	1
Generar acta de amonestación.	1

Tabla 15 Estimación de esfuerzo de la historias de usuario.

2.6.2 Plan de Iteraciones.

Todo proyecto que siga la metodología XP se ha de dividir en iteraciones de aproximadamente tres semanas. Una iteración es un período de tiempo en el que se realizan un conjunto de funcionalidades determinadas, que en el caso de la metodología aplicada corresponden a un grupo de historias de usuarios. Para determinar la cantidad de iteraciones se debe tener en cuenta la velocidad del proyecto, la cual es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de ellas.

Para maximizar el valor de negocio de cada historia de usuario, el cliente es quien decide qué historias se implementarán en cada iteración. Después de haber estimado el esfuerzo de cada historia de usuario, quedan distribuidas para cada iteración de la siguiente manera:

Iteración 1: En esta iteración será creada una gran parte de la arquitectura del sistema. Para lograr esto se implementan las historias de usuario que tienen mayor prioridad, entre las que se encuentran: Gestionar trabajador, Reportar evaluación mensual, Cargar modelos, Realizar búsquedas filtradas, Exportar a formatos estándares como pdf, xls, doc, html, sql y xml las planillas, modelos, actas e imprimirlas de forma directa, Controlar asistencia diaria mediante un scanner de barras. Al final de la iteración se tendrá un prototipo del sistema capaz de dar respuesta a algunas funcionalidades definidas. Se debe tener en cuenta que si existe alguna historia de usuario cuya implementación sea sencilla se le dedica el tiempo restante a otra más compleja.

Capítulo II. Características del Sistema

Iteración 2: En la segunda iteración se tienen las historias de usuario de mediana complejidad para su implementación, ellas son: Controlar capacitación, estudios y sanciones de los trabajadores, Registrar sanciones administrativas, Generar acta de admisión, Generar boleta de admisión, Generar modelo de baja.

Iteración 3: En esta iteración se implementan las historias de usuario Generar acta de advertencia, Generar certificado medico, Generar acta de amonestación y Controlar cumpleaños.

Número de la iteración.	Historias de usuario según la prioridad.	Duración en semanas.
1	Gestionar trabajador Reportar evaluación mensual Realizar búsquedas filtradas Exportar a formatos estándares como pdf, xls, doc, html, sql y xml las planillas, modelos, actas e imprimirlas de forma directa Controlar asistencia diaria mediante un scanner de barras Cargar modelos	5
2	Controlar capacitación, estudios y sanciones de los trabajadores Generar acta de admisión Generar boleta de admisión Generar modelo de baja	5
3	Generar acta de advertencia Generar certificado médico Generar acta de amonestación Controlar cumpleaños	3

Capítulo II. Características del Sistema

Tabla 16 Plan de iteración.

2.6.3 Plan de entrega.

En la realización del plan de entrega para la implementación se agruparon las historias de usuario en módulos que presenta el sistema según sus características funcionales.

Formas.	Historias de usuario.
Base de Datos	<ul style="list-style-type: none">• Gestionar trabajador• Controlar asistencia diaria mediante scanner• Exportar a formatos estándares como pdf, xls, doc, html, sql y xml las planillas, modelos, actas e imprimirlas de forma directa• Realizar búsquedas filtradas
Modelos	<ul style="list-style-type: none">• Cargar modelo• Acta de admisión• Boleta de admisión• Acta de advertencia• Modelo de baja• Evaluación docente• Acta Consejo de Dirección• Evaluación mensual• Modelo de asistencia• Acta de amonestación• Certificado médico

Tabla 17 Historias de usuario por módulos.

Capítulo II. Características del Sistema

Con el plan de entrega se puede llevar un control de las fechas donde serán entregadas las versiones del producto. Se les da seguimiento a las nuevas versiones creadas por algunos módulos que pueden afectar a otros.

Formas	1ra iteración	2da iteración	3ra iteración
Base de Datos	Inicio:30/01/2011 Fin: 3/03/2011		
Modelos		Inicio:3/03/2011 Fin:3/04/2011	Inicio:3/04/2011 Fin:28/05/2011

Tabla 18 Plan de entrega.

2.7 Conclusiones.

En este capítulo se trataron las principales características funcionales y no funcionales del sistema propuesto, recogidas en la lista de reserva del producto. Además se desarrollaron las fases de Exploración y Planificación según la metodología XP, destacando como principales artefactos las historias de usuario, plan de iteraciones y plan de entrega. Se determinó de las historias de usuario la estimación de esfuerzo, teniendo en cuenta la preferencia en el negocio y el riesgo en desarrollo, además de ser priorizadas y ubicadas en la iteración que le corresponde según su complejidad para la implementación.

Capítulo III. Construcción de la solución propuesta y Pruebas

3.1 Introducción.

En este capítulo se describen las fases de diseño e implementación de la metodología de desarrollo XP. Se especifican las tarjetas Clase-Responsabilidad-Colaboración (CRC) y las tareas de la ingeniería como principales artefactos que se obtienen en estas fases respectivamente. Además se detallan las iteraciones realizadas durante la etapa de implementación y se hace referencia al diseño de la base de datos del sistema

3.2 Diseño.

La metodología XP sugiere que se debe diseñar de la forma más simple y sencilla posible para lograr que la aplicación sea entendible para su implementación. El diseño simple se basa en la filosofía de que el mayor valor de negocio es entregado por el programa más sencillo que cumpla los requerimientos. Se enfoca en proporcionar un sistema que cubra las necesidades inmediatas del cliente. Este proceso permite eliminar redundancias y rejuvenecer los diseños obsoletos de forma sencilla (28).

3.2.1 Tarjetas CRC

Para el diseño de las aplicaciones la metodología XP propone el uso de tarjetas CRC como alternativa a los diagramas Lenguaje Unificado de Modelado (UML) de las clases, permiten al programador centrarse y apreciar el desarrollo orientado a objetos. En ellas se plasman las responsabilidades que tienen cada objeto, la clase a la cual pertenece y las clases que colaboran con cada responsabilidad.

A continuación se presentan las tarjetas CRC para el diseño de la aplicación.

<i>Base de Datos</i>	
<i>Responsabilidades</i>	<i>Colaboraciones</i>

Capítulo III. Construcción de la solución Propuesta y Pruebas

Insertar trabajador	FAdd
Modificar trabajador	FAdd
Eliminar trabajador	FAdd
Asistencia mediante un lector de código de barras	FAsistencia
Exportar a formatos estándares como pdf, xls, doc, html, sql y xml las planillas, modelos, actas e imprimirlas de forma directa	FReporte
Realizar búsquedas filtradas	Form1

Tabla 19 Tarjeta CRC de Base de Datos.

Modelos	
Responsabilidades	Colaboraciones
Cargar modelo	Form1
Acta de admisión	FActaAdmisión
Boleta de admisión	FBoletaAdmisión
Acta de advertencia	FAdvertencia

Capítulo III. Construcción de la solución Propuesta y Pruebas

Modelo de baja	FModeloBaja
Evaluación docente	FModeloEvaDocente
Acta Consejo de dirección	FActaConsDir
Evaluación mensual	FMoldeEvalMensual
Modelo de asistencia	FModeloAsistencia
Acta de amonestación	FActaAmonestación
Certificado médico	FCertificadoMédico

Tabla 20 Tarjeta CRC de modelos.

Para apoyar la técnica de las tarjetas CRC se presenta el diagrama de las clases persistentes y el modelo de datos. De esta forma se ayuda a comprender mejor la estructura y funcionamiento del sistema.

Capítulo III. Construcción de la solución Propuesta y Pruebas

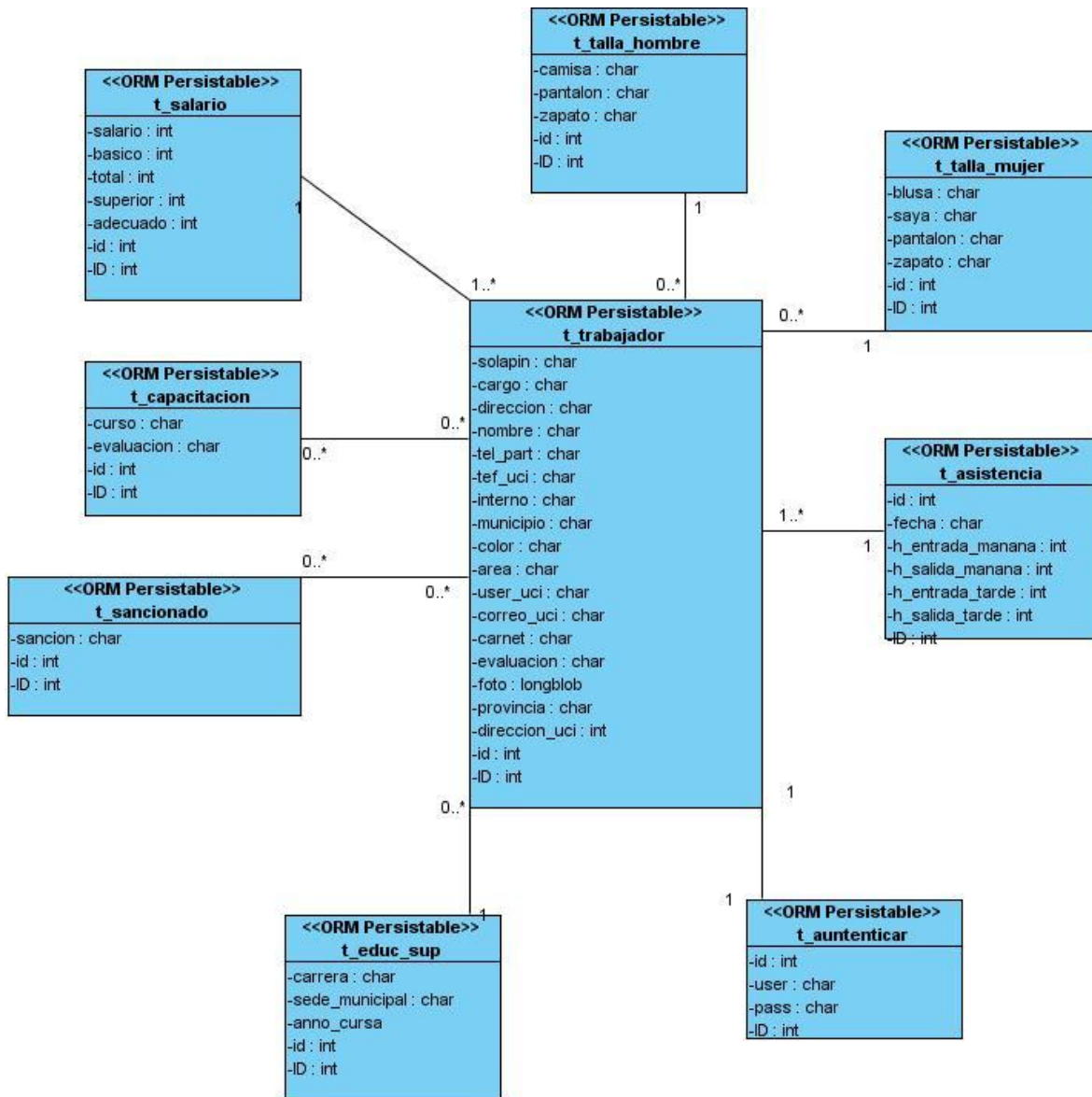


Figura 2. Diagrama de Clases Persistentes

Capítulo III. Construcción de la solución Propuesta y Pruebas

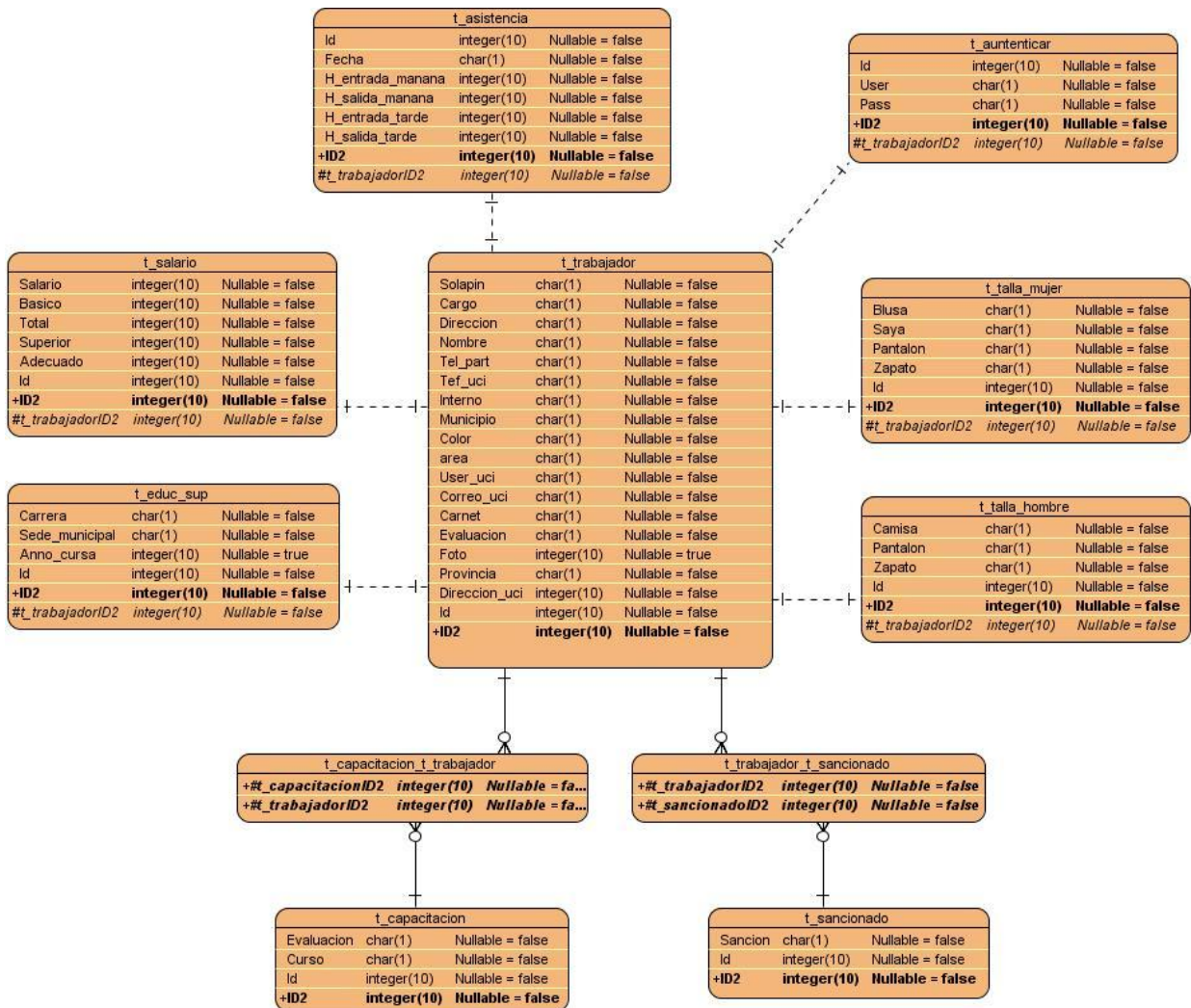


Figura 3. Modelo de Datos

Capítulo III. Construcción de la solución

Propuesta y Pruebas

3.2.2 Arquitectura en tres capas.

La arquitectura tres capas proporciona una buena organización y estructuración entre los distintos niveles de abstracción, donde, un cambio en uno de estos niveles no debe proporcionar cambios en los restantes. Las capas que la componen son (29):

Capa de presentación: es la que ve el usuario, presenta el sistema al mismo, le comunica la información y captura los datos que el usuario introduce en la aplicación. Esta capa se comunica únicamente con la del negocio enviando peticiones a la misma. También es conocida como interfaz gráfica y debe tener la característica de ser entendible y fácil de usar para el usuario.

Capa de negocio: es donde residen los programas que se ejecutan durante la ejecución de la aplicación. Tratan las peticiones del usuario y envían las respuestas tras el proceso. En ella se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la de presentación, para recibir las peticiones y presentar los resultados, y con la de acceso a datos, para interactuar con la base de datos, consultando y modificando sus informaciones.

Capa de acceso a datos: es donde residen los datos y la encargada de acceder y modificar a los mismos. Está formada por uno o más sistemas de gestión de bases de datos que realizan todo el almacenamiento de informaciones y que reciben solicitudes de almacenamiento o recuperación de datos desde la capa de negocio.

Ventajas:

- Brinda una mayor flexibilidad de desarrollo y de elección de plataformas sobre la cual montar las aplicaciones.
- Provee escalabilidad horizontal y vertical.
- Se mantiene la independencia entre el código de la aplicación (reglas y conocimiento del negocio) y los datos, mejorando la portabilidad de las aplicaciones.

Capítulo III. Construcción de la solución Propuesta y Pruebas

- Los lenguajes sobre los cuales se desarrollan las aplicaciones son estándares lo que hace más exportables las aplicaciones entre plataformas.
- Dado que mejora el rendimiento al optimizar el flujo de información entre componentes, permite construir sistemas críticos de alta fiabilidad.
- Disminuye el número de usuarios conectados a la base de datos.

Se selecciona el patrón arquitectónico Tres Capas para el diseño de la aplicación ya que sus ventajas se adaptan perfectamente a los objetivos que se quieren lograr, con esta arquitectura se desarrollan aplicaciones de manera clara y transparente, con la ventaja de poder desarrollar varias capas al mismo tiempo. El no estar atado a un lenguaje de programación, garantiza una fácil adaptación. Permite una centralización de los aspectos de seguridad, no replicación de lógica de negocio en los clientes permitiendo que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento.



Figura 4. Arquitectura en tres capas.

Capítulo III. Construcción de la solución

Propuesta y Pruebas

3.3 Implementación.

La presencia del cliente en las diferentes fases de XP como parte del equipo de trabajo es indispensable. A la hora de codificar o implementar una historia de usuario es aún más necesaria su presencia pues son ellos los que crean las historias de usuario, negocian los tiempos en los que serán implementadas, especifican detalladamente lo que esta hará y verifican que la historia implementada cumple la funcionalidad especificada cuando se realicen las pruebas. La codificación debe hacerse atendiendo a estándares de codificación ya creados. Los programadores lo han de seguir de tal manera que el código en el sistema se vea como si hubiera estado escrito por una sola persona. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

3.3.1 Tareas de implementación para cada módulo.

Tareas de implementación del módulo Base de Datos

<i>Tarea</i>	
Número de la tarea: 1	Historia de usuario: Gestionar trabajador.
Nombre de la tarea: Insertar trabajador.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.25
Programador responsable: Alejandro Luis Hernández Armas	
Descripción: El sistema muestra la opción para que el administrador pueda crear un trabajador, modificar un trabajador y eliminar un trabajador.	

Tabla 21 Tarea 1 de implementación. Insertar trabajador.

Capítulo III. Construcción de la solución Propuesta y Pruebas

Tarea	
Número de la tarea: 2	Historia de usuario: Gestionar trabajador.
Nombre de la tarea: Modificar trabajador.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Programador responsable: Idalmis Rodríguez Ramírez	
Descripción: El sistema muestra la opción para que el administrador pueda modificar los datos de un trabajador.	

Tabla 22 Tarea 2 de implementación. Modificar trabajador.

Tarea	
Número de la tarea: 3	Historia de usuario: Gestionar trabajador.
Nombre de la tarea: Eliminar trabajador.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Programador responsable: Alejandro Luis Hernández Armas	
Descripción: El sistema muestra la opción para que el administrador pueda eliminar un trabajador.	

Tabla 23 Tarea 3 de implementación. Eliminar trabajador.

Tarea	
Número de la tarea: 4	Historia de usuario: Controlar asistencia diaria mediante un lector de código de barras

Capítulo III. Construcción de la solución Propuesta y Pruebas

Nombre de la tarea: Controlar asistencia	
Tipo de tarea: Desarrollo	Puntos estimados: 0.33
Programador responsable: Alejandro Luis Hernández Armas	
Descripción: El sistema muestra la opción al administrador de controlara la asistencia diaria mediante un scanner	

Tabla 24 Tarea 4 de implementación. Controlar asistencia.

Tarea	
Número de la tarea: 5	Historia de usuario: Exportar a formatos estándares como pdf, xls, doc, html, sql y xml las planillas, modelos, actas e imprimirlas de forma directa
Nombre de la tarea: Exportar a formatos estándares	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
Programador responsable: Alejandro Luis Hernández Armas Idalmis Rodríguez Ramírez	
Descripción: El sistema muestra la opción de exportar a diferentes formatos estándares	

Tabla 25 Tarea 5 de implementación. Exportar a formatos estándares.

Tarea	
Número de la tarea: 6	Historia de usuario: Realizar búsquedas filtradas
Nombre de la tarea: Realizar búsquedas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50

Capítulo III. Construcción de la solución Propuesta y Pruebas

<p>Programador responsable: Alejandro Luis Hernández Armas Idalmis Rodríguez Ramírez</p>
<p>Descripción: El sistema muestra la opción de realizar las diferentes búsquedas necesarias</p>

Tabla 26 Tarea 6 de implementación. Realizar búsquedas filtradas.

Tareas de implementación para el Módulo Modelos

Tarea	
Número de la tarea: 7	Historia de usuario: Cargar modelo
Nombre de la tarea: Cargar modelo	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
<p>Programador responsable: Alejandro Luis Hernández Armas Idalmis Rodríguez Ramírez</p>	
<p>Descripción: El sistema muestra la opción de cargar los diferentes modelos que sean necesarios</p>	

Tabla 27 Tarea 7 de implementación. Cargar modelos.

Tarea	
Número de la tarea: 8	Historia de usuario: Acta de admisión
Nombre de la tarea: Generar acta de admisión	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
<p>Programador responsable: Alejandro Luis Hernández Armas</p>	

Capítulo III. Construcción de la solución Propuesta y Pruebas

Idalmis Rodríguez Ramírez
Descripción: El sistema muestra la opción de generar el acta de admisión de un trabajador

Tabla 28 Tarea 8 de implementación. Mostrar acta de admisión.

Tarea	
Número de la tarea: 9	Historia de usuario: Boleta de admisión
Nombre de la tarea: Generar boleta de admisión	
Tipo de tarea: Desarrollo	Puntos estimados: 0.33
Programador Responsable: Alejandro Luis Hernández Armas Idalmis Rodríguez Ramírez	
Descripción: El sistema brinda la posibilidad de generar la boleta de admisión de un trabajador	

Tabla 29 Tarea 9 de implementación. Mostrar boleta de admisión.

Tarea	
Número de la tarea: 10	Historia de usuario: Acta de advertencia
Nombre de la tarea: Generar acta de advertencia	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Programador Responsable: Alejandro Luis Hernández Armas Idalmis Rodríguez Ramírez	

Capítulo III. Construcción de la solución Propuesta y Pruebas

Descripción: El sistema muestra la opción de generar el acta de advertencia
--

Tabla 30 Tarea 10 de implementación. Generara acta de advertencia.

Tarea	
Número de la tarea: 11	Historia de usuario: Modelo de baja
Nombre de la tarea: Generar modelo de baja	
Tipo de tarea: Desarrollo	Puntos estimados: 0.33
Programador Responsable: Alejandro Luis Hernández Armas Idalmis Rodríguez Ramírez	
Descripción: El sistema muestra la opción de generar el modelo de baja de un trabajador.	

Tabla 31 Tarea 11 de implementación. Generar modelo de baja.

Tarea	
Número de la tarea: 12	Historia de usuario: Evaluación Docente
Nombre de la tarea: Generar evaluación docente	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
Programador Responsable: Alejandro Luis Hernández Armas Idalmis Rodríguez Ramírez	
Descripción: El sistema permitirá generar la evaluación docente de cada trabajador	

Tabla 32 Tarea 12 de implementación. Generar evaluación.

Capítulo III. Construcción de la solución Propuesta y Pruebas

Tarea	
Número de la tarea: 13	Historia de usuario: Acta Consejo de Dirección
Nombre de la tarea: Generar acta de consejo de dirección	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
Programador Responsable: Alejandro Luis Hernández Armas Idalmis Rodríguez Ramírez	
Descripción: El sistema brindará la posibilidad de generar el acta del consejo de dirección	

Tabla 33 Tarea 13 de implementación. Generar acta de consejo de dirección.

Tarea	
Número de la tarea: 14	Historia de usuario: Evaluación mensual
Nombre de la tarea: Generar evaluación mensual	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
Programador Responsable: Alejandro Luis Hernández Armas Idalmis Rodríguez Ramírez	
Descripción: El sistema debe brindar la posibilidad de generar la evaluación mensual de un trabajador	

Tabla 34 Tarea 14 de implementación. Generar evaluación mensual.

Tarea	
Número de la tarea: 15	Historia de usuario: Modelo de asistencia

Capítulo III. Construcción de la solución Propuesta y Pruebas

Nombre de la tarea: Generar modelo de asistencia	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
Programador Responsable: Alejandro Luis Hernández Armas Idalmis Rodríguez Ramírez	
Descripción: El sistema brindará la posibilidad de generar el modelo de asistencia	

Tabla 35 Tarea 15 de implementación. Generar modelo de asistencia.

Tarea	
Número de la tarea: 16	Historia de usuario: Acta de amonestación
Nombre de la tarea: Generar acta de amonestación	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
Programador Responsable: Alejandro Luis Hernández Armas Idalmis Rodríguez Ramírez	
Descripción: El sistema brindará la posibilidad de generar un acta de amonestación	

Tabla 36 Tarea 16 de implementación. Generar acta de amonestación.

Tarea	
Número de la tarea: 17	Historia de usuario: Certificado médico
Nombre de la tarea: Generar certificado médico	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50

Capítulo III. Construcción de la solución Propuesta y Pruebas

Programador Responsable: Alejandro Luis Hernández Armas Idalmis Rodríguez Ramírez
Descripción: El sistema brindará la posibilidad de generar un certificado médico

Tabla 37 Tarea 17 de implementación. Generar certificado médico.

3.4 Pruebas.

3.4.1 Introducción.

En el presente capítulo se explica dónde encajan las pruebas del sistema dentro de la clasificación de las pruebas que para ello establece la metodología XP. Dicha metodología propone las pruebas de aceptación. Estas pruebas las realiza el cliente y son verificadas en cada entrega del producto donde se chequea la planificación establecida.

3.4.2 Pruebas de aceptación.

Uno de los pilares de la Programación Extrema es el uso de pruebas para comprobar el funcionamiento de los códigos que se han implementado. La metodología alienta a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados en la realización de modificaciones y refactorizaciones.

XP divide las pruebas del sistema en dos grupos: las pruebas unitarias, encargadas de verificar el código, diseñada por los programadores y las pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad diseñada por el cliente. Estas últimas son más importantes que las pruebas unitarias dado que significan la satisfacción del cliente con el producto desarrollado, el final de una iteración y el comienzo de la siguiente. Las pruebas de aceptación sirven para

Capítulo III. Construcción de la solución Propuesta y Pruebas

evaluar las distintas tareas en las que ha sido dividida una historia de usuario y asegurar su correcto funcionamiento.

A continuación se mencionan las pruebas de aceptación de las funcionalidades realizadas al sistema:

- Pruebas de aceptación para la historia de usuario Gestionar trabajadores.
- Pruebas de aceptación para la historia de usuario Controlar asistencia diaria mediante scanner.
- Pruebas de aceptación para la historia de usuario Exportar a formatos estándares como pdf, xls, doc, html, sql y xml las planillas, modelos, actas e imprimirlas de forma directa.
- Pruebas de aceptación para la historia de usuario Realizar búsquedas filtradas.
- Pruebas de aceptación para la historia de usuario Cargar modelo.
- Pruebas de aceptación para la historia de usuario Acta de admisión.
- Pruebas de aceptación para la historia de usuario Boleta de admisión.
- Pruebas de aceptación para la historia de usuario Acta de advertencia.
- Pruebas de aceptación para la historia de usuario Modelo de baja.
- Pruebas de aceptación para la historia de usuario Evaluación docente.
- Pruebas de aceptación para la historia de usuario Acta consejo de dirección.
- Pruebas de aceptación para la historia de usuario Evaluación mensual.
- Pruebas de aceptación para la historia de usuario Modelo de asistencia.
- Pruebas de aceptación para la historia de usuario Acta de amonestación.
- Pruebas de aceptación para la historia de usuario Certificado médico.

Capítulo III. Construcción de la solución Propuesta y Pruebas

3.4.3 Pruebas de aceptación para cada historia de usuario.

Pruebas de aceptación para la historia de usuario Gestionar trabajadores

Caso de Prueba de Aceptación.	
Código: T1-HU1	Historia de usuario: Gestionar trabajadores.
Nombre : Insertar trabajador	
Descripción: Prueba para verificar si se ha creado una cuenta de usuario en el sistema.	
Condiciones de Ejecución: Para crear una cuenta de trabajador debe de autenticarse en el sistema como administrador	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección “Base de Datos” • Seleccionar la opción “Gestionar trabajadores” • Llenar los campos necesarios. • Pulsar la opción de adicionar trabajador 	
Resultado Esperado: La cuenta de trabajador ha sido creada	
Evaluación de la Prueba: Satisfactoria	

Tabla 38 Caso de Prueba de aceptación 1. Insertar trabajador.

Caso de Prueba de Aceptación.	
Código: T2-HU1	Historia de usuario: Gestionar trabajador
Nombre : Modificar trabajador	
Descripción: Prueba para verificar si se ha modificado un trabajador	

Capítulo III. Construcción de la solución Propuesta y Pruebas

Condiciones de Ejecución: Para modificar una cuenta de trabajador debe de autenticarse en el sistema como administrador
Entrada / Pasos de ejecución: <ul style="list-style-type: none">• Entrar en la sección “Base de Datos”• Seleccionar la opción “Gestionar trabajadores”• Llenar los campos necesarios• Pulsar la opción que se desee modificar en el trabajador
Resultado Esperado: La cuenta de trabajador ha sido modificada
Evaluación de la Prueba: Satisfactoria

Tabla 39 Caso de Prueba de aceptación 2 Modificar trabajador.

Capítulo III. Construcción de la solución Propuesta y Pruebas

Caso de Prueba de Aceptación.	
Código: T3-HU1	Historia de usuario: Gestionar trabajador
Nombre : Eliminar trabajador	
Descripción: Prueba para verificar si se ha eliminado un trabajador	
Condiciones de Ejecución: Para eliminar una cuenta de trabajador debe autenticarse en el sistema como administrador	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección “Base de Datos” • Seleccionar la opción “Gestionar trabajadores” • Seleccionar un trabajador • Pulsar la opción de eliminar 	
Resultado Esperado: La cuenta de trabajador ha sido eliminada	
Evaluación de la Prueba: Satisfactoria	

Tabla 40 Caso de Prueba de aceptación 3. Eliminar trabajador.

Caso de Prueba de Aceptación.	
Código: T1-HU2	Historia de usuario: Controlar capacitación
Nombre : Controlar capacitación	
Descripción: Prueba para comprobar la capacitación de cada trabajador o sea los cursos que tienen vencidos y los que se le asignarán además de la evaluación obtenida en los mismos	
Condiciones de Ejecución: Para controlar la capacitación de los trabajadores se debe estar autenticado en el sistema como administrador	
Entrada / Pasos de ejecución:	

Capítulo III. Construcción de la solución Propuesta y Pruebas

<ul style="list-style-type: none"> • Entrar en la sección “Base de Datos” • Seleccionar la opción “Gestionar trabajadores” • Seleccionar un trabajador
Resultado Esperado: La capacitación ha sido controlada satisfactoriamente
Evaluación de la Prueba: Satisfactoria

Tabla 41 Caso de Prueba de aceptación 4. Controlar capacitación.

Caso de Prueba de Aceptación.	
Código: T2-HU2	Historia de usuario: Controlar sanciones de los trabajadores
Nombre : Controlar sanciones de los trabajadores	
Descripción: Prueba para comprobar si un trabajador ha sido sancionado y los motivos por los cuales fue sancionado	
Condiciones de Ejecución: Para controlar las sanciones a trabajadores se debe estar autenticado en el sistema como administrador	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección “Base de Datos” • Seleccionar la opción “Gestionar trabajadores” • Seleccionar un trabajador 	
Resultado Esperado: El requisito controlar sanciones de los trabajadores ha sido implementado correctamente	
Evaluación de la Prueba: Satisfactoria	

Tabla 42 Caso de Prueba de aceptación 5. Controlar sanciones de los trabajadores.

Capítulo III. Construcción de la solución Propuesta y Pruebas

Caso de Prueba de Aceptación.	
Código: T1-HU3	Historia de usuario: Reportar evaluación mensual
Nombre: Evaluación Docente	
Descripción: Prueba para verificar las evaluaciones de los trabajadores docentes	
Condiciones de Ejecución: Para realizar la acción de reportar evaluación de un trabajador docente mensual se debe de estar autenticado en el sistema como administrador	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección “Modelos” • Seleccionar la opción “Evaluación docente” 	
Resultado Esperado: La evaluación docente ha sido realizada	
Evaluación de la Prueba: Satisfactoria	

Tabla 43 Caso de Prueba de aceptación 6. Evaluación Docente.

Caso de Prueba de Aceptación.	
Código: T2-HU3	Historia de usuario: Reportar evaluación mensual
Nombre: Evaluación Trabajador	
Descripción: Prueba para verificar la evaluación de un trabajador	
Condiciones de Ejecución: Para realizar la evaluación de un trabajador debe de autenticarse en el sistema como administrador	

Capítulo III. Construcción de la solución Propuesta y Pruebas

Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar en la sección Base de Datos • Seleccionar “Gestionar trabajadores”
Resultado Esperado: La evaluación del trabajador ha sido realizada
Evaluación de la Prueba: Satisfactoria

Tabla 44 Caso de Prueba de aceptación 7. Evaluación trabajador.

Caso de Prueba de Aceptación.	
Código: T1-HU4	Historia de usuario: Cargar modelos
Nombre : Cargar modelos	
Descripción: Prueba para comprobar la carga de los diferentes modelos que sean necesarios	
Condiciones de Ejecución: Para realizar la carga de los modelos debe autenticarse en el sistema como administrador	
<ul style="list-style-type: none"> • Entrada / Pasos de ejecución: • Entrar a la sección Modelos • Pulsar la opción “Cargar modelos” 	
Resultado Esperado: La operación de cargar los modelos ha sido ejecutada satisfactoriamente.	
Evaluación de la Prueba: Satisfactoria	

Tabla 45 Caso de Prueba de aceptación 8. Cargar modelos.

Capítulo III. Construcción de la solución Propuesta y Pruebas

3.5 Conclusiones.

La fase de diseño en la metodología XP genera como artefacto fundamental las tarjetas CRC las cuales permiten a los programadores centrarse y apreciar el desarrollo orientado a objetos garantizando de esta manera una mejor calidad del código en la fase de implementación. Con las tareas de la ingeniería se le dio cumplimiento a las funcionalidades del sistema recogidas en las historias de usuario. Conjuntamente, en el presente capítulo se hizo una breve descripción de la fase de Pruebas, haciendo referencia a las pruebas de aceptación del producto las cuales verifican si el cliente está satisfecho con el software elaborado y permiten además comprobar el correcto funcionamiento del sistema. Se abordó la importancia de las pruebas en la metodología XP las cuales ayudan a identificar y corregir fallos u omisiones que favorecen la comunicación entre el cliente y el equipo de desarrollo.

Conclusiones generales

Luego de la investigación realizada y con el objetivo de apoyar el proceso de gestión de información de Recursos Humanos de la Dirección de Producción Audiovisual del centro FORTES se arribó a las siguientes conclusiones:

- Pudo constatar que ninguna de las aplicaciones existentes resuelven el problema o pueden ser utilizadas para resolver la situación existente en el Departamento de RRHH de la Dirección de Producción Audiovisual.
- Se logró implementar un sistema de gestión de información que facilita a la asistente del Departamento de Producción Audiovisual la gestión de los recursos humanos.
- Las pruebas realizadas al sistema permitieron corroborar que este cumple con las funcionalidades previstas.

Recomendaciones

Una vez cumplido el objetivo general propuesto se recomienda:

- Poner en práctica la propuesta realizada y hacer extensible el sistema a otras áreas.
- Agregar nuevas funcionalidades según las necesidades que vayan surgiendo en el transcurso del tiempo.