

# **Universidad de las Ciencias Informáticas**

## **Facultad 1**



### **Título: Elaboración de una aplicación que diagnostique el grado de acercamiento de los proyectos productivos al Programa de Mejora de la UCI.**

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

#### **Autores:**

Elisandra Soria Silva  
Arianne Prades Roque

#### **Tutores:**

Ing. Yusleydi Fernández del Monte  
Ing. Sonia Guerrero Lambert

**CIUDAD DE LA HABANA, JULIO DEL 2009.**

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

Firma del Autor

Elisandra Soria Silva

\_\_\_\_\_

Firma del Autor

Arianne Prades Roque

\_\_\_\_\_

Firma del Autor

Ing. Yusleydi Fernández del Monte

\_\_\_\_\_

Firma del Tutor

Ing. Sonia Guerrero Lambert

## DATOS DE CONTACTO

Ing. Yusleydi Fernández del Monte

- Graduada en el año 2008 de Ingeniero en Ciencias Informáticas.
- Tiene 4 años de experiencia en el trabajo vinculado con la Calidad de Software.
- Tiene 1 año de experiencia en el trabajo vinculado con la Gestión de Conocimientos.
- Tiene 1 año de experiencia en la docencia universitaria.
- Ha publicado y realizado presentaciones en eventos sobre temas de gestión de conocimientos y calidad de software.
- Se desempeña como profesora de la facultad 10

Ing. Sonia Guerrero Lambert

- Graduada en el año 2008 de Ingeniero en Ciencias Informáticas.
- Tiene 4 años de experiencia en el trabajo vinculado con la Calidad de Software.
- Tiene 1 año de experiencia en el trabajo vinculado con la Gestión de Conocimientos.
- Tiene 1 año de experiencia en la docencia universitaria.
- Ha publicado y realizado presentaciones en eventos sobre temas de gestión de conocimientos y calidad de software.
- Se desempeña como profesora de la facultad 10

## AGRADECIMIENTOS

Le agradezco a la nuestra Revolución por permitirme este gran sueño de ser una profesional y sobre todo a Fidel. A mis tutoras por ayudarme en este trabajo de diploma y por tener paciencia y por todo lo que han hecho con nosotras. A todos los que nos han ayudado en el transcurso de la elaboración de la tesis como Marlon, Marino y en especial a Félix por su ayuda incondicional.

Pienso que en los agradecimientos no se para mencionar a las personas que de una forma u otra aportaron su granito de arena, es válido mencionar a todas aquellas que se involucraron conmigo en estos 5 años de carrera como a mis amigos, mis verdaderos amigos que me han apoyado y han sabido aceptarme con mis virtudes y defectos, a ellos no los voy a mencionar saben perfectamente quienes son, aquí están incluidos Liván y Nery que aparte de ser profesores son mis amigos también.

Les agradezco a todos por venir.

## DEDICATORIA

Le dedico este trabajo de diploma a mis compañeros que vienen desde la vocacional conmigo, a todos mis amigos de Santiago que no están aquí conmigo como a Yorley porque siempre tuvo confianza en mí y me daba aliento cada vez que lo necesite.

También se la dedico a mi toda mi familia sin excepción de ninguno. A mi familia de la habana por acogerme como una más, mi familia de arroyo y en especial a 2 personitas especiales que viven San Miguel del Padrón por cuidarme, guiarme y quererme en estos 5 años, ellos son mi tía Aracelis y a Félix. A mis hermanos a los 4 que tengo, Elo te adoro.

Pero en especial se la dedico a 2 seres que quiero que son mis padres por darme el mayor regalo que todos tenemos que es la vida, por cuidarme en estos 23 años, por ayudarme emocional y monetariamente, por darme su amor incondicional, por ayudarme a ser cada día mejor y más completa como persona.

A mi mamita que ha sido madre y padre en una etapa de mi vida, te quiero dar las gracias por ser mi amiga, hermana y compañera, por hacer de mí una mujer plena y darme tu apoyo siempre.

Dedico este trabajo a las personas que día a día me colman de cariño, a todos mis amigos de la UCI que ayudaron a que llegara a ser una persona mejor y marcaron una huella indeleble en mi paso por la universidad, en especial a Raúl, Yenelis, Yudalis, Carlitos y Noema por haberme SOPORTADO todos estos años, nunca los olvidaré.

A mi familia de Centro Habana, gracias por su apoyo incondicional. A mis padres, a mi abuelo, a mi hermano, a mis tíos y primos.

## RESUMEN

La mejora de procesos ha tomado un papel determinante en la identificación, integración, medición de las buenas prácticas existentes en la organización y desarrollo de software. La UCI como universidad productiva, trabaja en una propuesta de solución para los proyectos productivos que ayude a mejorar la calidad y establezca como deben funcionar los proyectos. El presente trabajo es la automatización de esta propuesta de solución que tiene como objetivo la elaboración de una aplicación para ayudar a la organización y reajuste en el desarrollo de las actividades de los proyectos productivos en la UCI. Para ello se utilizaron diferentes métodos de investigación como métodos empíricos y teóricos. Como resultado se obtuvo una aplicación capaz de evaluar (en porciento) el grado de acercamiento de los proyectos productivos al programa de mejora establecido por la Dirección Central de Calidad. La aplicación permite generar reportes de los proyectos evaluados para que puedan mejorar, adicionar o eliminar variables según la propuesta de solución predeterminada hasta el momento.

Después de validar la propuesta planteada se demostró que cumple con las expectativas del cliente. Es una aplicación eficiente, muy útil y permite obtener de forma sencilla y rápida el resultado de la evaluación de los proyectos.

## PALABRAS CLAVE

**programa de mejora, mejora de procesos, evaluación de proyectos.**

## TABLA DE CONTENIDOS

AGRADECIMIENTOS .....	2
DEDICATORIA.....	3
RESUMEN.....	4
TABLA DE CONTENIDOS .....	5
INDICE DE TABLAS .....	6
INDICE DE FIGURAS.....	8
INTRODUCCIÓN .....	9
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	14
1.3 Variables que determinan el programa de mejora continua en la UCI .....	16
1.4 Teoría sobre el algoritmo.....	25
1.5 Metodologías de desarrollo de software .....	26
1.6 Aplicaciones de escritorio.....	28
1.7 Importancia de crear una aplicación para diagnosticar el grado de acercamiento al programa de mejora en la UCI.....	31
CAPITULO 2. CARACTERISTICAS DEL SISTEMA .....	32
Descripción de la aplicación propuesta .....	32
2.1 Requisitos no funcionales.....	33
2.2 Algoritmo .....	33
2.3 Roles de Usuarios.....	36
2.4 Descripción del proceso de construcción .....	37
2.5 Fase de exploración.....	39
2.5.1 Historias de Usuario .....	39
2.6 Fase de Planificación.....	49
2.6.1 Estimación de esfuerzo por historia de usuario .....	49
2.6.2 Plan de Iteraciones.....	50
2.6.2.1 Iteración 1 .....	50
2.6.2.2 Iteración 2 .....	50
2.6.2.3 Iteración 3 .....	51
2.6.3 Plan de duración de las iteraciones.....	51
2.6 Plan de entregas .....	52
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS.....	53
3.1 Primera Iteración.....	53
3.2 Segunda Iteración .....	59
3.3 Tercera Iteración .....	67
3.4 Pruebas.....	68
3.5 Diseño de la Aplicación .....	75
CONCLUSIONES.....	78
RECOMENDACIONES.....	79
BIBLIOGRAFÍA.....	80
ANEXOS .....	82
GLOSARIO DE TERMINOS .....	87

## INDICE DE TABLAS

<b>Tabla 1.1: Roles y Responsabilidades.</b>	17
<b>Tabla 2.1: Descripción de los Actores.</b>	36
<b>Tabla 2.2: Historia de usuario “Autenticar”.</b>	40
<b>Tabla 2.3: Historia de usuario “Evaluar proyecto”.</b>	41
<b>Tabla 2.4: Historia de usuario “Adicionar variable”.</b>	42
<b>Tabla 2.5: Historia de usuario “Cambiar contraseña”.</b>	43
<b>Tabla 2.6: Historia de usuario “Modificar variable”.</b>	44
<b>Tabla 2.7: Historia de usuario “Listar proyectos”.</b>	45
<b>Tabla 2.8: Historia de usuario “Generar reporte”.</b>	46
<b>Tabla 2.9: Historia de usuario “Eliminar Proyecto”.</b>	47
<b>Tabla 2.10: Historia de usuario “Guardar Proyecto”.</b>	48
<b>Tabla 2.11: Historia de usuario “Cargar Proyecto”.</b>	49
<b>Tabla 2.12: Estimación de esfuerzo por historias de usuario.</b>	50
<b>Tabla 2.13: Plan de Iteraciones.</b>	51
<b>Tabla 2.14: Plan de entregas</b>	52
<b>Tabla 3.1: Historia de Usuario Evaluar proyecto con Estimación y Tiempo Real</b>	53
<b>Tabla 3.2: Tarea de Ingeniería Nro.1 para la Historia de Usuario Evaluar Proyecto.</b>	54
<b>Tabla 3.3: Tarea de Ingeniería Nro.2 para la Historia de Usuario Evaluar Proyecto.</b>	55
<b>Tabla 3.4: Tarea de Ingeniería Nro.3 para la Historia de Usuario Evaluar Proyecto.</b>	56
<b>Tabla 3.5: Tarea de Ingeniería Nro.4 para la Historia de Usuario Evaluar Proyecto.</b>	57
<b>Tabla 3.6: Historia de Usuario Autenticar con su semanas de Estimación y Tiempo Real</b>	57
<b>Tabla 3.7: Tarea de Ingeniería Nro.1 para la Historia de Usuario Autenticar.</b>	58
<b>Tabla 3.8: Tarea de Ingeniería Nro.2 para la Historia de Usuario Autenticar.</b>	58
<b>Tabla 3.9: Historia de Usuario Adicionar Variable con sus semanas de Estimación y Tiempo Real</b>	59
<b>Tabla 3.10: Tarea de Ingeniería Nro.1 para la Historia de Usuario Adicionar Variable.</b>	59
<b>Tabla 3.12: Tarea de Ingeniería Nro.1 para la Historia de Usuario Modificar Variable.</b>	60
<b>Tabla 3.17: Tarea de Ingeniería Nro.2 para la Historia de Usuario Modificar Variable.</b>	61
<b>Tabla 3.18: Tarea de Ingeniería Nro.3 para la Historia de Usuario Modificar Variable.</b>	62
<b>Tabla 3.19: Tarea de Ingeniería Nro.4 para la Historia de Usuario Modificar Variable.</b>	62
<b>Tabla 3.20: Historia de Usuario Listar proyecto con sus semanas de Estimación y Tiempo Real</b>	63
<b>Tabla 3.21: Tarea de Ingeniería Nro.1 para la Historia de Usuario Listar proyecto.</b>	63
<b>Tabla 3.22: Historia de Usuario Generar reporte con sus semanas de Estimación y Tiempo Real</b>	63
<b>Tabla 3.23: Tarea de Ingeniería Nro.1 para la Historia de Usuario Generar reporte.</b>	64
<b>Tabla 3.24: Historia de Usuario Eliminar proyecto con sus semanas de Estimación y Tiempo Real</b>	64
<b>Tabla 3.25: Tarea de Ingeniería Nro.1 para la Historia de Usuario Eliminar proyecto.</b>	65
<b>Tabla 3.26: Historia de Usuario Guardar proyecto con sus semanas de Estimación y Tiempo Real</b>	65
<b>Tabla 3.27: Tarea de Ingeniería Nro.1 para la Historia de Usuario Guardar proyecto.</b>	66
<b>Tabla 3.28: Historia de Usuario Cargar proyecto con sus semanas de Estimación y Tiempo Real</b>	66
<b>Tabla 3.29: Tarea de Ingeniería Nro.1 para la Historia de Usuario Cargar proyecto.</b>	67
<b>Tabla 3.30: Historia de Usuario Cambiar contraseña con sus semanas de Estimación y Tiempo Real</b>	67



<b>Tabla 3.31: Tarea de Ingeniería Nro.1 para la Historia de Usuario Cambiar contraseña.</b>	68
<b>Tabla 3.32: Caso de Prueba de Aceptación HU1P1.</b>	69
<b>Tabla 3.33: Caso de Prueba de Aceptación HU2P1.</b>	69
<b>Tabla 3.34: Caso de Prueba de Aceptación HU3P1.</b>	70
<b>Tabla 3.35: Caso de Prueba de Aceptación HU4P1.</b>	70
<b>Tabla 3.36: Caso de Prueba de Aceptación HU5P1.</b>	71
<b>Tabla 3.37: Caso de Prueba de Aceptación HU5P2.</b>	71
<b>Tabla 3.38: Caso de Prueba de Aceptación HU6P1.</b>	73
<b>Tabla 3.39: Caso de Prueba de Aceptación HU7P1.</b>	73
<b>Tabla 3.40: Caso de Prueba de Aceptación HU8P1.</b>	74
<b>Tabla 3.41: Caso de Prueba de Aceptación HU9P1.</b>	74
<b>Tabla 3.42: Caso de Prueba de Aceptación HU10P1.</b>	75

## INDICE DE FIGURAS

<b>Fig. 1 Comparación de las metodologías de desarrollo según Baird.....</b>	<b>27</b>
<b>Fig. 2 Mapa conceptual “Diagrama del producto”.....</b>	<b>32</b>
<b>Fig. 3 Ejemplo de variables para calcular el algoritmo.....</b>	<b>36</b>
<b>Fig. 4 Mapa conceptual “Roles y Funcionalidades”.....</b>	<b>37</b>
<b>Fig. 5 Mapa conceptual “Diagrama de proceso”.....</b>	<b>38</b>
<b>Fig. 6 Formulario “Presentación de la aplicación”.....</b>	<b>76</b>
<b>Fig. 7 Formulario “Funcionalidades de la aplicación”.....</b>	<b>76</b>
<b>Fig. 8 Formulario “Evaluación de Proyectos”.....</b>	<b>77</b>
<b>Fig. 9 Formulario “Adicionar y Eliminar Variables”.....</b>	<b>77</b>

## INTRODUCCIÓN

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. Es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. Es medible y varía de un sistema a otro o de un programa a otro. Es la aptitud de un producto o servicio para satisfacer las necesidades del usuario. Es decir: “la calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”. (IEEE, Std. 610-1990).

Se verificó la gran necesidad de buscar y adaptar modelos de procesos que permitan incrementar la calidad de los proyectos. La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

La calidad en los proyectos es de gran importancia ya que los beneficios de aplicarla no solo repercuten en el área de desarrollo de los proyectos, sino que también se dejan notar en el resto de las áreas de negocio y en el cliente final.

La mejora de los procesos significa optimizar la efectividad y la eficiencia mejorando también los controles, reforzando los mecanismos internos para responder a las contingencias y las demandas de nuevos y futuros clientes. La mejora de procesos ha tomado un papel determinante en la identificación, integración, medición de las buenas prácticas existentes en la organización y desarrollo de software. La mejora de procesos intenta cambiar la forma en que la gente ejecuta las actividades para satisfacer mejor los objetivos del negocio.

- ✓ Mejorar implica siempre cambiar.
- ✓ La mejora se debe definir en términos de objetivos de negocio y se debe manejar del mismo modo.

Para ayudar a las empresas en la mejora de sus procesos, se crearon los modelos de mejora de procesos, ya que un modelo de mejora de procesos favorece que la organización "ponga sobre la mesa" sus procesos actuales, reflexionen sobre ellos para comprender qué es lo que hace y por qué lo hace, y en base a este estudio los optimicen para que estos sean lo más "eficaces" y "eficientes" como sea posible. La importancia que logra tener esta técnica es que a través de su aplicación se contribuye a mejorar las debilidades y hacer que la organización se fortalezca. Con la mejora continua en las organizaciones se logra que se desarrollen sus procesos de una forma más productiva y eficiente para así reducir costos y poder ofrecer un producto o servicio de calidad.

La industria cubana del software trabaja arduamente en lograr su desarrollo, por lo que la mejora de sus procesos es una meta inmediata a cumplir. La UCI (Universidad de las Ciencias Informáticas), como impulsora y protagonista del desarrollo ha comenzado a investigar en cómo aplicar programas de mejora de procesos en los proyectos productivos. La UCI decide usar las fortalezas que brindan algunos modelos para lograr la calidad de los procesos que conforman los sistemas de software que se producen en la institución, debido a que la calidad de los mismos depende de los procesos usados para desarrollarlos.

Con el propósito de asegurar la calidad y establecer cómo deberán funcionar los proyectos, en la UCI se aplicó el programa de mejora continua a 5 de los 25 polos existentes. Para llevar a cabo este programa se establece una propuesta de solución por parte de Alfonso Alva Rosano, Presidente del Tecnológico de Monterrey y el SIE Center (EIS por sus siglas en inglés, es un sistema de información para directivos), ambos de México. El elevado costo de esta solución impidió que se pudiese implantar en los polos restantes; esto provoca que se detenga el establecimiento del programa de mejora, y por ende el proceso de implantación del nivel 2 de CMMI (Capability Maturity Model Integrated), trayendo como consecuencias que:

- ✓ Se pierdan posibilidades de mercado.
- ✓ Se vende un producto que no cuenta con una óptima calidad.
- ✓ No se valoran los productos elaborados y servicios prestados por la Universidad.
- ✓ Los productos no cuentan con un modelo de calidad certificado mundialmente.

Esto impide que la Universidad se amplíe como productora de software a nivel global y logre un mayor desarrollo económico del país, lo cual constituye uno de los objetivos principales del centro.

La UCI está definiendo programas de mejora para los 20 polos restantes, pero se desconoce si este programa de mejora establecido por el SIE Center es compatible con alguno de estos proyectos. Si existen proyectos que puedan utilizar este programa de mejora, esto ayudaría a un ahorro de tiempo, esfuerzo y personal trabajando en la definición y adaptabilidad de nuevos procesos.

Constituyéndose como **problema científico** lo siguiente:

¿Cómo elaborar una aplicación que ayude a diagnosticar el grado de acercamiento de proyectos productivos de la UCI al proceso de mejora continua?

Teniendo como **objetivo de la investigación** elaborar una aplicación que ayude a diagnosticar el grado de acercamiento de proyectos productivos de la UCI al proceso de mejora continua.

El **objeto de estudio** del trabajo de diploma es el Proceso de desarrollo de software.

Enfocando el **campo de acción** en el Desarrollo de aplicaciones de escritorio que determinen el grado de acercamiento de proyectos de la UCI al programa de mejora continua.

Para darle cumplimiento al objetivo de la investigación se proponen las siguientes **tareas de la investigación**:

- ✓ Estudiar técnicas matemáticas para determinar el grado de acercamiento de proyectos de la UCI al programa de mejora continua.
- ✓ Investigar las metodologías que se utilizan en el proceso de desarrollo de software para escoger la más propicia.
- ✓ Estudiar lenguajes de programación utilizados para desarrollar aplicaciones de escritorio para seleccionar el lenguaje adecuado en la implementación del sistema.
- ✓ Diseñar una aplicación con las funcionalidades especificadas por el cliente para obtener una visión detallada del software a construir.
- ✓ Implementar una aplicación que diagnostique el grado de acercamiento de proyectos productivos de la UCI al programa de mejora.
- ✓ Instalar la aplicación en el entorno del cliente para obtener el criterio de aceptación del mismo.

Se toma como guía en el desarrollo de la tesis las siguientes **preguntas científicas**:

¿Qué variables permiten conocer hasta qué punto está inmerso un proyecto en el Programa de Mejora de Procesos en la UCI?

¿Qué técnicas matemáticas permiten hallar el grado de acercamiento de proyectos de la UCI al programa de mejora continua?

¿Cuáles son las metodologías y lenguajes de programación que se utilizan en el proceso de desarrollo de software?

En la realización de este trabajo se utilizan los siguientes **Métodos Científicos**:

- **Métodos teóricos**

- ✓ Analítico-Sintético: analiza las teorías planteadas, documentos, planteamientos, entre otros, para luego extraer los elementos más importantes que se relacionan con el objeto de estudio y adaptarlo a la situación.
- ✓ Inductivo-Deductivo: Se hace uso de deducciones para llegar a tener una visión clara de lo que se quiere hacer y adquirir así nuevos conocimientos. Este método se aplica en inducción y deducción de los lenguajes de programación que se va a escoger.
- ✓ Modelación: permite crear abstracciones con el objetivo de investigar la realidad y además posibilita conocer la respuesta de los procesos sin tener que ejecutar los mismos en el mundo real. Permite modelar en la fase de análisis y diseño cómo debe quedar el producto final.

- **Métodos empíricos**

- ✓ Entrevista (ver anexo 1): con el propósito de conocer las funcionalidades que debe tener el sistema a desarrollar teniendo en cuenta las necesidades del cliente. Se usa al inicio y durante todo el desarrollo del trabajo, pues a medida que se desarrolla el sistema se analiza si el mismo cumple las expectativas requeridas.

Para el desarrollo de la aplicación se definen III capítulos. En el capítulo I se trata sobre los objetivos del programa de mejora, los conceptos definidos por el cliente, las variables que determinan en el programa

de mejora, la metodología de desarrollo y el lenguaje de programación. En el capítulo II se aborda sobre el algoritmo a utilizar en la aplicación, las historias de usuario a utilizar para guiarse en la implementación de la aplicación, la Fase de Planificación y el Plan de Iteraciones. En el capítulo III se realiza la implementación correspondiente a todas las historias de usuario abordadas en el capítulo II y las pruebas relacionadas a cada historia de usuario.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

En este capítulo se detallan las definiciones referentes al Programa de Mejora de Procesos en la UCI. Se examinan los materiales existentes como vía de apoyo a la propuesta de solución. Se establecen las variables que determinan el programa de mejora continua. Se escoge el lenguaje y metodología más conveniente a utilizar en la aplicación. Para finalizar se brinda la propuesta de solución acordada según las autoras.

### **1.1 Variable**

Una variable es un símbolo que representa un elemento no especificado de un conjunto dado. Dicho conjunto es llamado conjunto universal de la variable, universo o dominio de la variable, y cada elemento del conjunto es un valor de la variable.

Según la influencia que se asigne a unas variables sobre otras, podrán ser:

- ✓ Variables independientes: Son las que el investigador escoge para establecer agrupaciones en el estudio, clasificando intrínsecamente a los casos del mismo. Un tipo especial son las variables que modifican al resto de las variables independientes y que de no tenerse en cuenta adecuadamente pueden alterar los resultados por medio de un sesgo.
- ✓ Variables dependientes: Son las variables de respuesta que se observan en el estudio y que podrían estar influenciadas por los valores de las variables independientes.

La Dirección Central de Calidad define como variable: los aspectos a medir que se van a tener en cuenta cuando se va a evaluar los proyectos productivos en la UCI.



## 1.2 Programa de mejora continua en la UCI

Un programa de mejora continua es una secuencia de instrucciones para llevar a cabo la mejora continua a los proyectos productivos. Con la misma se consigue lograr una mayor calidad y eficiencia en dichos proyectos.

En la actualidad existen modelos de procesos de mejora continua entre los que se encuentran: Six Sigma, PDCA, CMM, IDEAL, ISO/IEC 12 207, CMMI entre otros. CMMI (Capability Maturity Model Integrated) es un modelo de mejora de procesos organizado en áreas de proceso que determinan el nivel de capacidad o madurez de una organización. La Universidad de las Ciencias Informáticas realizó un contrato con especialistas de calidad para llevar a cabo el programa de mejora continua en algunos proyectos de la Universidad.

El programa de mejora continua en la UCI fue firmado en Ciudad de La Habana, en abril del 2008, por Alfonso Alva Rosano (Presidente del Instituto de Tecnologías de Información del Tecnológico de Monterrey) y la UCI. En ese mismo año se firma el contrato y se inicia el trabajo en octubre.

Para mejorar los procesos de desarrollo de software de la UCI se lleva a cabo el proyecto de mejora bajo el modelo CMMI nivel 2 presentados por el SIE CENTER, que incluye:

- ✓ Evaluación inicial de los procesos actuales de producción de software de la UCI.
- ✓ Consultoría, implementación e institucionalización de los procesos de Nivel 2 de CMMI.
- ✓ Evaluación formal basada en el SCAMPI (Standard CMMI Appraisal Method for Process Improvement).

Entre los objetivos del servicio de CMMI en la UCI están:

- ✓ Ayudar a la UCI a revisar su estrategia de mejora de procesos de software, para asegurar que su organización está basada en procesos y con un programa de mejora continua alineado con sus objetivos de negocio.

- ✓ Ayudar a la UCI a establecer las bases y fundamentos para seguir mejorando sus procesos y fortalecer su cultura de calidad en el desarrollo de software.
- ✓ Alinear los procesos de desarrollo de software con los principios y requisitos del modelo CMMI, estableciendo planes de mejora con los que la organización oriente sus procesos hacia la consecución de sus metas.

### **1.3 Variables que determinan el programa de mejora continua en la UCI**

Luego de un estudio realizado por la Dirección Central de Calidad, se obtuvieron algunas variables que determinan que un proyecto este inmerso en el Programa de mejora de procesos en la UCI. Estas variables tienen prioridad (0, 1, 2), entre menor es su prioridad mayor es su importancia. A continuación se muestran las variables utilizadas en el proceso:

- ✓ Tipos de proyectos según sus características: Declara los tipos de proyectos existentes.
  - Proyectos clásicos: Los proyectos clásicos comienzan con el levantamiento de los requerimientos utilizando diversas técnicas como pueden ser: modelado de negocio, ingeniería inversa, investigación entre otras. Pueden llegar hasta la etapa de despliegue de la aplicación o incluir el soporte de software durante un tiempo limitado.
  - Proyectos de mantenimiento: Los proyectos de mantenimiento no llevan una etapa de modelado de negocio y comienzan directamente en la descripción de los requerimientos. Al igual que los proyectos clásicos pueden concluir en el despliegue o pueden incluir el soporte con un tiempo limitado.
- ✓ Ciclo de vida de los proyectos: Establece los ciclos de vida asociados a los proyectos pilotos del Programa de Mejora.

Para los proyectos del alcance del Programa de mejora se define el siguiente ciclo de vida básico:

- Modelado del Negocio (no se ejecuta para los proyectos de mantenimiento).

- Requerimientos.
- Análisis y Diseño.
- Implementación.
- Pruebas.
- Despliegue.
- Soporte. (es opcional para todos los proyectos y su tiempo debe ser limitado)

Nota: El tiempo de soporte debe ser de tiempo limitado.

Para los proyectos de Mantenimiento no se ejecuta el Modelado de negocio.

La etapa de soporte es opcional para todos los proyectos.

✓ Roles y responsabilidades: Define la responsabilidad asociada a cada rol en el proyecto.

**Tabla 1.1: Roles y Responsabilidades.**

Rol	Responsabilidades
Asegurador de la calidad	<p>Elabora el plan de gestión de la calidad.</p> <p>Elabora el plan de mediciones.</p> <p>Elabora los planes de prueba.</p> <p>Participa en las revisiones técnicas formales de los artefactos.</p> <p>Participa en las revisiones con el cliente de los entregables.</p> <p>Guía el diseño y ejecución de las pruebas internas.</p> <p>Participa en el análisis y recolección de los datos para las mediciones.</p> <p>Vela por el cumplimiento de las políticas de la organización y reglas bases del proyecto.</p> <p>Colabora en las auditorías que se les realicen al proyecto.</p> <p>Coordina y colaborar con las pruebas de liberación externa al proyecto.</p> <p>Crea una cultura de calidad en el proyecto.</p>
Administrador de	<p>Identifica los elementos de configuración.</p> <p>Establece las líneas base del proyecto.</p>

configuración	<p>Mantiene las diferentes librerías del proyecto.</p> <p>Mantiene el control de las versiones de los artefactos del proyecto.</p> <p>Provee el mecanismo administrativo para precipitar, preparar, evaluar y aprobar o reprobar el procesamiento de propuestas de cambio.</p> <p>Genera métricas de la configuración del sistema.</p> <p>Configura y administrar las herramientas para la Gestión de la Configuración.</p> <p>Proporciona las herramientas para el desarrollo del proyecto.</p> <p>Genera los entregables de la aplicación.</p> <p>Elabora plan de gestión de la configuración.</p> <p>Audita la gestión de configuración dentro del proyecto.</p>
Alta Gerencia	<p>Jefes de polo.</p> <p>Vicedecanos de producción.</p> <p>Decano.</p> <p>Dirección General de Producción.</p>
Analista	<p>Participa con el cliente y el usuario final recogiendo las entradas de los stakeholders.</p> <p>Captura los requisitos y definir las prioridades.</p> <p>Lleva a cabo las actividades del análisis.</p> <p>Realiza la especificación de requisitos.</p> <p>Desarrolla el modelo de análisis del sistema.</p> <p>Documenta el flujo de análisis.</p> <p>Realiza el seguimiento de los requerimientos durante todo el desarrollo del proyecto.</p> <p>Participa en el diseño de la solución.</p> <p>Diseña las pruebas.</p>
Arquitecto de información	<p>Identifica la visión, misión y objetivos del producto, equilibrando las necesidades de la organización patrocinadora y la de su público.</p> <p>Realiza el estudio de homólogos para conocer el estado del arte del producto que se quiere desarrollar.</p> <p>Realiza auditoría de información identificando las entidades de</p>

	<p>recursos de información conociéndose como: servicios, fuentes, sistema, contenidos.</p> <p>Realiza la organización y representación de los contenidos a través de: definición de la taxonomía, diseño del sistema de navegación y diseño del sistema de etiquetado para el sistema de navegación.</p> <p>Realiza diagramación de diagramas tipos.</p>
<p>Arquitecto de software</p>	<p>Define todos los elementos bases de la arquitectura del proyecto.</p> <p>Identifica todos los posibles escenarios de despliegue de la aplicación</p> <p>Identifica componentes horizontales de la aplicación.</p> <p>Determina de conjunto con los diseñadores las interfaces de integración tanto internas como externas.</p> <p>Elabora el documento de arquitectura para el expediente de proyecto.</p> <p>Define las herramientas, bibliotecas, componentes, Frameworks y otros componentes que permitan acelerar y mejorar el trabajo del proyecto.</p> <p>Define de conjunto con el jefe de proyecto el flujo de desarrollo basado en las herramientas identificadas.</p> <p>Vela por el cumplimiento de los requerimientos de hardware.</p> <p>Planifica la integración de los componentes del sistema.</p>
<p>Diseñador de BD</p>	<p>Define la lógica de la base de datos.</p> <p>Encargado de modelar e implementar la base de datos del sistema.</p> <p>Define el gestor de base de datos a usar.</p> <p>Define la herramienta de modelado para bases de datos relacionales.</p> <p>Define las políticas de cambio sobre los elementos de datos. Define los algoritmos de réplica, sincronización, respaldo, recuperación de la base de datos.</p> <p>Define las políticas de almacenamiento de los datos.</p>

	<p>Define las políticas de uso de los diferentes objetos de bases de datos ante situaciones particulares.</p> <p>Usa el diagrama entidad-relación para generar el diseño físico de la base de datos.</p> <p>Crea y mantiene el ambiente de la base de datos para el funcionamiento de la aplicación. Interviene en el ajuste del desempeño de la aplicación.</p> <p>Administra gestor de BD</p> <p>Configura roles y usuarios</p> <p>Administra BD.</p>
Diseñador gráfico	<p>Encargado de realizar todo el diseño gráfico que requiera el sistema.</p> <p>Interviene en la creación del prototipo.</p> <p>Define las pautas para el diseño de la interfaz.</p>
Diseñador de pruebas	<p>Participa en la confección de la estrategia y el plan de pruebas.</p> <p>Identifica los métodos, las técnicas, herramientas y directrices apropiadas para implementar las pruebas necesarias.</p> <p>Establece en ambiente de comprobación</p> <p>Encargado de diseñar casos de pruebas para el sistema</p> <p>Dirige la definición del enfoque de prueba y garantiza la implementación satisfactoria.</p> <p>Diseña las pruebas.</p> <p>Genera los casos y datos de prueba.</p> <p>Puede fungir como probador.</p> <p>Analiza y evalúa el estado del producto de trabajo comprobado.</p> <p>Evalúa el desempeño del probador y la calidad de las pruebas realizadas.</p>
Diseñador de sistema	<p>Traduce los requerimientos identificados por los analistas hacia la tecnología utilizada para desarrollar en el proyecto.</p> <p>Vela que se cumplan todas las orientaciones del arquitecto con respecto al flujo de trabajo y las herramientas a utilizar.</p>

		<p>Participa en la concepción del sistema a desarrollar de conjunto con los analistas y arquitectos.</p> <p>En dependencia del tamaño del proyecto también podría encargarse del diseño de la BD sino esta tarea la llevaría otra persona.</p>
Económico		<p>Estima costos del proyecto.</p> <p>Realiza análisis de factibilidad del proyecto.</p> <p>Gestiona presupuesto del proyecto.</p>
Especialista funcional		<p>Asesora al proyecto en cuanto a los procesos del negocio que se automatizarán.</p>
Gestor de conocimiento		<p>Gestiona la capacitación interna del proyecto.</p> <p>Gestiona la formación desde la producción.</p> <p>Acredita competencias.</p> <p>Gestiona las lecciones aprendidas.</p>
Implantador de soluciones		<p>Coordina la capacitación del personal interno y externo del proyecto tanto como para tareas de desarrollo como para tareas de despliegue.</p> <p>Controla el estado técnico del equipamiento asignado al proyecto.</p> <p>Capacitación de clientes en los productos desarrollados.</p> <p>Atiende las solicitudes del cliente en la etapa de despliegue.</p> <p>Realiza el levantamiento tecnológico de los centros a informatizar para ver las necesidades para el despliegue.</p>
Ingeniero de procesos		<p>Realiza el diagnóstico y evaluación de la organización o área que se va a informatizar.</p> <p>Identifica posibles áreas de mejora.</p> <p>Determina y diseñar los procesos de la organización o área que se va a informatizar.</p> <p>Evalúa y propone mejora al propio proceso de desarrollo de software que está llevando el proyecto.</p>

<p>Jefe de proyecto</p>	<p>Administra recursos          Controla Recursos          Monitorea la adherencia a procesos.          Participa en las RTF.          Participa en las revisiones con el cliente de los entregables.          Administra la capacitación interna al proyecto.          Participa en la elaboración de la visión general del proyecto.          Guía el proceso de identificación y mitigación de los riesgos.          Evalúa a los miembros del proyecto según su desempeño.          Desarrolla el plan de proyecto o de desarrollo de software.          Gestiona las interacciones con clientes y usuarios.</p>
<p>Planificador</p>	<p>Elabora y controla cronogramas del proyecto.          Planifica y gestiona los recursos del proyecto.</p>
<p>Probador</p>	<p>Encargado de seguir los planes de pruebas.          Ejecuta los casos de prueba y genera no conformidades asociadas al mismo.          Registra los resultados de las pruebas.          Analiza los resultados de las pruebas realizadas.</p>
<p>Programador</p>	<p>Convierte la especificación del sistema en código fuente ejecutable.          Desarrolla el diseño teniendo en cuenta la arquitectura.          Elabora las pruebas de unidad.          Desarrolla el prototipo de la interfaz de usuario.          Integra los componentes que forman parte de la solución.          Ejecuta los casos de prueba y generar no conformidades asociadas al mismo.          Registra y analiza los resultados de las pruebas.</p>



Otras variables que son en forma de preguntas en la aplicación:

✓ Esquema de comunicación.

La comunicación entre los grupos de trabajo se establece a través de un contacto previamente establecido, cada grupo designa la persona (preferentemente el líder del grupo) que establecerá el único canal de comunicación entre los grupos de trabajo. Debe constar en acta o como acuerdo de una minuta del grupo la persona que servirá de contacto entre grupos. Se actualizará el presente documento ante cada cambio.

✓ Mecanismo de comunicación con consultores.

El envío de la documentación se realizará vía correo electrónico que incluirá un archivo compactado de la entrega. Debe enviarse un documento con el contenido de la entrega.

Las entregas se efectuarán en las fechas convenidas en el cronograma.

Para las entregas se utilizará el formato “entrega #<número de la entrega>\_<fecha de la entrega>” sin las comillas. Ej. Entrega #01\_081202.

✓ Mecanismo para registro de sugerencias.

Las sugerencias podrán ser realizadas de manera directa solo por los miembros del MSG, EGP (engineer process group), TWG (time work group) y QA.

✓ Mecanismos de comunicación con la organización.

Define las actividades que se realizarán para mantener informada a la organización sobre el avance del programa de mejora.

✓ Registro de documentos.

Mantiene el registro de los documentos creados.

✓ Cantera para TWGs v1.0.

Crea una cantera para la selección de las personas que formarán parte de los TWGs por cada paquete de mejora.

- ✓ Plan de Capacitación de PPQA.  
Plan de capacitación y formación del personal necesario para desarrollar el proceso de PPQA en la Universidad de las Ciencias Informáticas (UCI).
  
- ✓ Plantilla de Listas de Verificación.  
Define las revisiones en la fase de implementación a todos los Proyectos Productivos de la Universidad de las Ciencias Informáticas.
  
- ✓ IPP: Procesos definidos para la Infraestructura productiva de la UCI. [1]
  - Administración de requisitos.
    - Planifica la administración de requisitos.
    - Determina el proveedor de requisitos.
    - Prepara el entorno de administración de requisitos.
    - Planifica y prepara el entorno de la iteración.
    - Obtiene requisitos.
    - Especifica requisitos.
  - Entendimiento y compromiso.
    - Analiza los requisitos del cliente y el producto.
    - Analiza impacto de los requisitos.
    - Acepta requisitos del cliente.
    - Acepta requisitos del producto.
  - Control de cambios.
    - Registra la solicitud de cambio.
    - Traceo.
    - Evalúa el impacto de los cambios.
    - Registra y actualizar estado del cambio.
    - Entendimiento y compromiso.
    - Planea y asigna el cambio.
    - Actualiza los productos de trabajo.

---

<sup>1</sup> Este es un concepto tomado según los criterios del cliente.

- Traceo.
  - Crea la matriz de trazabilidad.
  - Analiza el efecto de la trazabilidad.
  - Actualiza la matriz de trazabilidad.
- Control de inconsistencias.
  - Revisa las inconsistencias.
  - Asigna responsable de resolver las inconsistencias.
  - Resuelve las inconsistencias.
  - Monitorea las inconsistencias hasta su cierre.
  - Acepta el cambio.
  - Traceo.
  - Reporta el estado del proceso de Administración de requisitos a la alta gerencia.
  - Describe las entidades que persisten en el proceso de definir el sistema.
  - Define las características de las salidas del sistema (reportes).
- Seguimiento/Escalamiento de NC.
  - Elaboración del Plan de acciones correctivas.
  - Monitorear no conformidades.
  - Verificar si se está ejecutando la acción correctiva.
  - Actualizar los Registros de no conformidades y acciones correctivas del proyecto.
  - Notificar al coordinador de PPQA la no conformidad a escalar.
  - Actualizar planes.
  - Especificar requisitos del producto.

#### **1.4 Teoría sobre el algoritmo**

"Tanto por ciento" significa que: por cada 100 unidades, se toman x unidades de esas 100, es decir dada una proporción entre una parte y el todo de algo (la suma de las partes), es establecer una proporción equivalente entre un porcentaje (la parte) y 100 (el todo).  $\text{Parte/todo} = \%/100$ .

Sirve para facilitar la comprensión de la importancia de esa parte en el todo, y para comparar situaciones similares con otras partes del mismo todo. Aclarar que dados los conceptos cuando se menciona unidades

son las variables. Se utiliza este método porque conociendo el porcentaje de los valores máximos de las variables correspondientes permiten conocer si el proyecto se adecua completamente al programa de mejora. Es necesario determinar el porcentaje o la parte que representa verdaderamente los resultados evaluados en la aplicación del algoritmo porque de esta manera se puede conocer realmente como se está comportando el proyecto.

### **1.5 Metodologías de desarrollo de software**

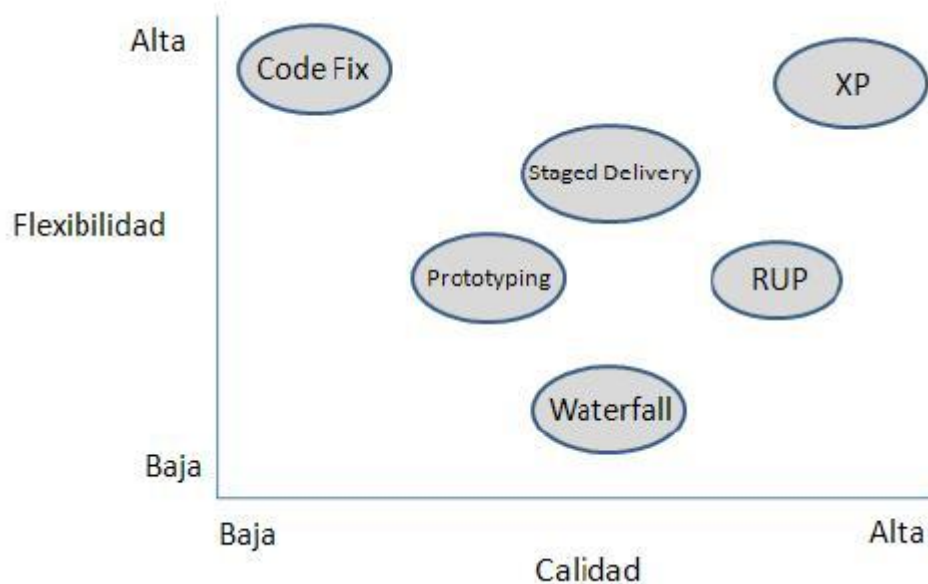
En la actualidad no existe una metodología general para desarrollar cualquier proyecto de desarrollo de software, sino que estas tienen prácticas específicas que las hacen más o menos óptimas, dependiendo de las características del proyecto. Se podría decir que en estos últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo del software, los llamados métodos pesados y los métodos ligeros. La diferencia fundamental entre ambos es que mientras los métodos pesados intentan conseguir el objetivo común por medio de orden y documentación, los métodos ligeros tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso. (Molpeceres, 2002)

Una de estas tendencias está representada por el Proceso Unificado de Desarrollo (del inglés Rational Unified Process o RUP), que es una metodología pesada y constituye uno de los procesos más generales de los existentes en la actualidad, ya que en realidad está pensado para adaptarse a cualquier proyecto. Se caracteriza por ser muy organizativo, se basa en roles y tiene una vasta documentación.

Por otro lado se encontró la Programación Extrema (del inglés Extreme Programming o XP), que pertenece a las metodologías ligeras (también conocida como ágiles) e intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. XP se caracteriza por desarrollar en pareja, mantener el mínimo de organización, el código pertenece a todos y es pobre en documentación. (Molpeceres, 2002)

Mientras que RUP intenta reducir la complejidad del software por medio de estructuras y la preparación de las tareas pendientes en función de los objetivos de la fase y actividad actual, XP, como toda metodología ágil, lo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. (Molpeceres, 2002)

A continuación se expone en la figura 1 diferentes metodologías, que son caracterizadas en una gráfica de flexibilidad contra calidad. (Baird, 2003)



**Fig. 1 Comparación de las metodologías de desarrollo según Baird.**

Para el desarrollo de la aplicación se escoge como metodología de desarrollo XP, ya que se ajusta en gran medida a la misma. Se basa en la realimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. Es adecuada especialmente para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico, es una metodología ágil especialmente pensada para proyectos en los cuales los equipos de desarrollo son pequeños, con plazos reducidos y basados en nuevas tecnologías. (Beck, 2002).

A continuación se muestran varios aspectos definidos en la programación extrema que influyeron en la decisión de utilizar esta metodología para el desarrollo del sistema. XP:

- ✓ Destacada en los procesos ágiles de desarrollo del software (desarrollo a corto plazo).
- ✓ Pone más énfasis en la adaptabilidad que en la previsibilidad.

- ✓ Considera los cambios de requisitos sobre la marcha como un aspecto natural, inevitable e incluso deseable en el desarrollo de proyectos.
- ✓ Recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto.
- ✓ Programación en parejas (cada miembro realiza una acción que el otro no está haciendo actualmente: mientras que uno codifica las pruebas de unidad el otro piensa en la clase que satisfaga la prueba, se sugiere a menudo que los dos socios cambien de papeles por lo menos cada media hora o después de que se haga una prueba de unidad).
- ✓ Desarrollo iterativo e incremental.
- ✓ Grupo pequeño y muy integrado (máximo 12 personas).
- ✓ Simplicidad (Propone el principio de hacer que la cosa más simple pueda funcionar, en relación al proceso y la codificación. Es mejor hacer hoy algo simple, que hacerlo complicado y probablemente nunca usarlo mañana).

## 1.6 Aplicaciones de escritorio

Las aplicaciones de escritorio se definen como aquellas que utiliza una empresa para proporcionar servicios de software a otras empresas de forma centralizada y a través de una red. Las mismas tienen muchas ventajas:

- ✓ Licencias: el cliente no paga licencias. Simplemente se suscribe a un servicio por el que paga mensualmente.
- ✓ Navegación e interfaz de usuario más rápida que en una aplicación web.
- ✓ Normalmente no es necesaria una conexión a internet ya que funciona localmente o en la intranet de la empresa.
- ✓ Es mejor opción cuando es necesario realizar numerosos cálculos o procesos de CPU muy intensivos
- ✓ Fácil acceso a recursos locales: disco duro, impresora, memoria de vídeo, sonido, ratón, teclado, etc.
- ✓ Fácil acceso a aplicaciones locales en caso de que sea necesario compartir datos entre aplicaciones.

Para la programación de estas se usan distintos lenguajes:

**Java** es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de

objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

El lenguaje de programación java tiene entre sus ventajas que permite realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras. Es independiente de la plataforma de desarrollo.

Entre sus desventajas existen diferentes tipos de soporte técnico para la misma herramienta, por lo que el análisis de la mejor opción se dificulta. Para manejo a bajo nivel deben usarse métodos nativos, lo que limita la portabilidad. Existen herramientas como el JBuilder que permiten generar interfaces gráficas de manera sencilla, pero tienen un costo adicional. Puede ser que no haya JDBC para bases de datos poco comerciales. Se tiene que descargar todo el applet, sólo la primera vez. Se ejecuta en el cliente, no el servidor. Necesita la instalación de una ENORME máquina virtual.

**C++** es un lenguaje de programación diseñado a mediados de los años 1980 tiene como características: Un lenguaje multiparadigma. Es un lenguaje híbrido y de fácil configuración. Bajo costo de mantenimiento y utilización. Su dispositivo hardware requerido es de fácil instalación, este puede ser interno o externo.

Consta con ventajas como ser un lenguaje de nivel intermedio, pudiéndose utilizar tanto para escribir software de bajo nivel, como drivers y componentes de sistemas operativos, como para el desarrollo rápido de aplicaciones, según el marco de trabajo con el que se disponga. Los compiladores de C++ generan código nativo con un alto grado de optimización en memoria y velocidad, lo que lo convierte en uno de los lenguajes más eficientes. Lenguaje de programación orientado a objetos. Lenguaje muy didáctico, gracias a este lenguaje puedes aprender muchos otros lenguajes con gran facilidad, como C#, Java, Visual Basic, Javascript, PHP, entre otros. Es muy potente en lo que se refiere a creación de sistemas complejos, un lenguaje muy robusto. Actualmente, puede compilar y ejecutar código de C, ya viene con librerías para realizar esta labor. Es un lenguaje muy empleado, existen muchos tutoriales en línea, libros, códigos fuentes abiertos, hay material de sobra y basta para aprender lo necesario y mucho más con este lenguaje. Existen muchos algoritmos cuyo pseudocódigo se encuentra ya desarrollado en

C++, de manera que puedes tomarlo y amoldarlo a tu solución (porque el que veas un fragmento de código no asegura que sea correcto al 100%).

Tiene como desventajas el uso de librerías dinámicas muy complejo. Elaborar un sistema en C++ es como construir un rascacielos: tiene buen soporte y es robusto, pero si existen errores en los pisos inferiores toda la parte superior se viene abajo terriblemente. No es recomendable para desarrollo de páginas Web. Baja velocidad de acceso. Requiere de un hardware y un software específico. Incompatibilidad de servicios. Sistema analógico.

**C Sharp** es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Tiene facilidad de uso (el ambiente de trabajo es muy cómodo ya que tiene un ambiente amigable y clásico de las aplicaciones de Windows). Programación orientada a objetos: esta forma de programación ahorra mucho código.

Presenta ventajas como declaraciones en el espacio de nombres al empezar a programar algo, se puede definir una o más clases dentro de un mismo espacio de nombres. Tipos de datos: en C# existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.

Para una mejor utilización del Visual Studio .NET este tiene como desventajas que debe portar siempre una versión reciente del mismo. Además el sistema debe contar con Windows NT 4 o superior, tener alrededor de 4 gigas de espacio libre para instalación.

Para implementar la aplicación se utiliza C Sharp, porque como lenguaje nativo de su famosa plataforma .Net para aplicaciones web y de escritorio, C# conjunta tanto aspectos de C++ como de Java y Visual Basic, pero de una forma más versátil y mejorada agregándole cada vez más elementos que faciliten su uso. El ambiente de trabajo es muy cómodo, tiene un ambiente amigable y clásico de las aplicaciones de Windows. En cuanto a la forma de programar, será fácil de usar para quien está familiarizado con C++, porque su estructuración básica es muy similar, sin embargo C# ahorra muchos pasos “tediosos” de otros lenguajes como la creación de funciones complejas desde cero y declaración de variables globales.



## **1.7 Importancia de crear una aplicación para diagnosticar el grado de acercamiento al programa de mejora en la UCI**

Luego de un estudio realizado a nivel mundial y en la universidad se comprobó que hasta el momento no existe ninguna aplicación capaz de diagnosticar el grado de acercamiento de los proyectos productivos al programa de mejora.

Una aplicación para conocer el grado de acercamiento de los proyectos productivos al programa de mejora continua es de importancia primordial, pues su objetivo es lograr una mayor eficiencia, calidad, y ahorrar tiempo en la entrega de los mismos. Sus posibles formas de uso van a proporcionar nuevas ideas para ajustar cada uno de los proyectos hacia la mejor vía de solución posible; permitiendo una reorientación sin pérdida de tiempo en la ardua tarea, lo que contribuye al máximo rendimiento del tiempo en el desarrollo de los proyectos.

### **Propuesta de solución al problema**

Se define hacer una aplicación que ayude a diagnosticar el grado de acercamiento de los proyectos al programa de mejora continua. Se decide utilizar para el desarrollo de la misma:

- ✓ C Sharp como lenguaje de programación.
- ✓ XP como metodología ágil.

## CAPITULO 2. CARACTERISTICAS DEL SISTEMA

En este capítulo se definen las características de la aplicación a desarrollar, haciendo referencia a la situación problemática que da origen a la misma. Se explican los principales artefactos de la metodología empleada y que debe tener la solución propuesta. A continuación se muestra un mapa conceptual que describe las funcionalidades del producto y los usuarios que interactúan con las mismas.

### Descripción de la aplicación propuesta

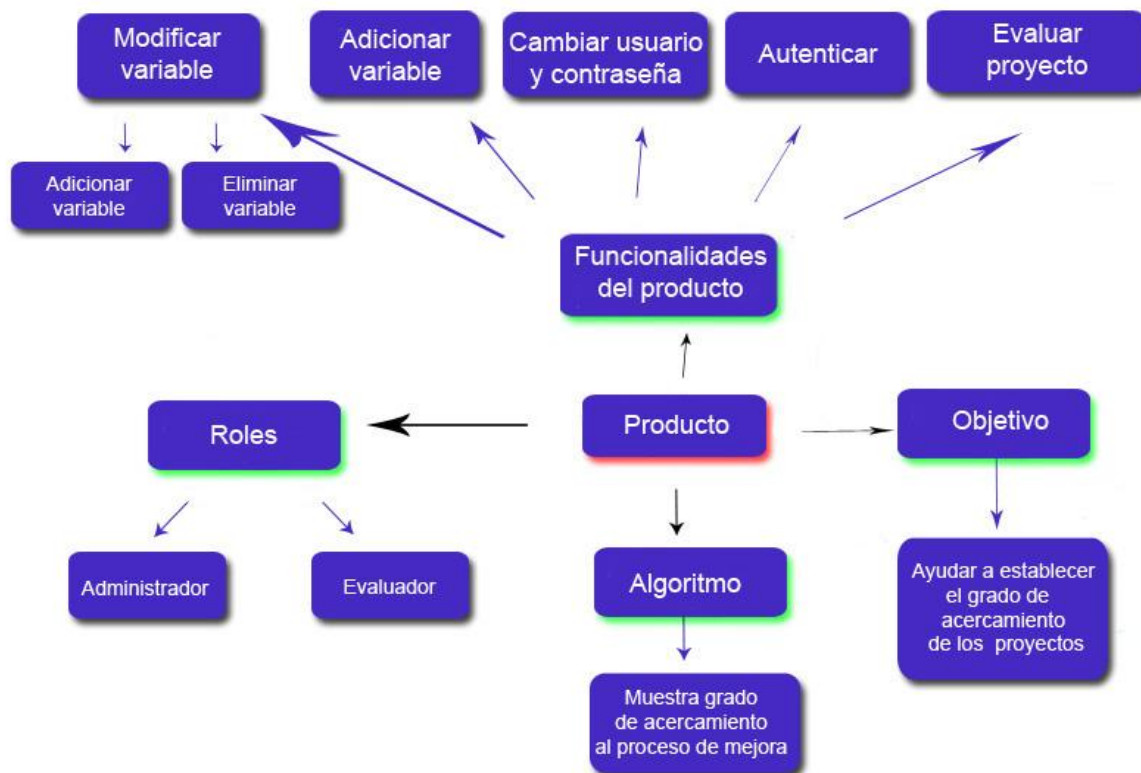


Fig. 2 Mapa conceptual "Diagrama del producto".

Este producto tiene como objetivo principal desarrollar una aplicación capaz de diagnosticar el grado de acercamiento de los proyectos productivos en la UCI al programa de mejora de procesos. Se considera la existencia de dos roles: el rol de administrador del sistema y el rol del evaluador cada uno con sus funcionalidades específicas. La aplicación cuenta además con un algoritmo encargado de calcular el porcentaje real de acercamiento de cada proyecto al programa de mejora.

## 2.1 Requisitos no funcionales

- ✓ Interfaz de usuario: El sistema tiene una interfaz sencilla y amigable para que pueda ser utilizada sin mucho entrenamiento por el usuario.
- ✓ Estaciones de trabajo (PC Cliente) Sistema operativo: Windows XP y Vista, Tarjeta de Red: 256 MB de RAM, Procesador: Pentium 4 (ó superior), Framework.Net 2.0 ó superior.
- ✓ Usabilidad: El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente de aplicación de escritorio en sentido general.
- ✓ Seguridad: Identifica al usuario antes de que pueda realizar cualquier acción sobre el sistema, protege la información manejada por el sistema de accesos no autorizados, garantiza que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo.
- ✓ Soporte: Se realizó pruebas al software para comprobar su funcionalidad.

## 2.2 Algoritmo

Las variables aparte de su prioridad (0, 1, 2) pueden ser simples o compuestas, tienen un valor máximo y uno real. El valor máximo ya lo tiene predeterminado. La sumatoria de los valores máximos de las variables de prioridad 0 representan un 60 %. La sumatoria de los valores máximos de las variables de prioridad 1 representa un 30%. La sumatoria de los valores máximos de las variables de prioridad 2 representa un 10%. Finalmente la suma de los valores máximos de todas las variables representa un 100%. A continuación se representa lo antes descrito.

$$V_a = \sum_1^n V_n \leq 60\%$$

$$V_b = \sum_1^n V_n \leq 30\%$$

$$V_c = \sum_1^n V_n \leq 10\%$$

$$V_T = \sum_a^c V_c \leq 100\%$$

Para saber el valor máximo de cada variable en específico, el algoritmo divide los valores máximos de cada prioridad entre la cantidad de variables que lo componen que pueden ser simples o compuestas. Ej. Si en la prioridad 0 se tienen 2 variables compuestas, el algoritmo divide 60/2 y cada una de estas variables tienen como valor máximo 30% y así sucesivamente hasta llegar a sus variables simples. Lo mismo ocurre con las demás prioridades.

Para obtener el valor real de cada variable se empieza con las simples hasta llegar a las compuestas porque el valor real de las variables compuestas se las da la suma de los valores reales de sus variables simples o compuestas es decir la que esté dentro de ella. En la aplicación, cuando se están evaluando los proyectos, en las variables simples se le pregunta al evaluador cómo es la forma en que tiene definida las variables el proyecto del usuario, el resultado puede tomar uno de los siguientes valores:

- ✓ No se realiza.
- ✓ Casi se realiza.
- ✓ Se realiza a medias.
- ✓ Se realiza casi completo.
- ✓ Se realiza completo.

Esta es la forma definida por la Dirección Central de Calidad para saber cualitativamente de qué forma está definida esta variable en los proyectos.

De acuerdo a lo que marque el evaluador se le dará a esta variable evaluada su valor real, tomando así valores cuantitativos como se muestran a continuación:

- ✓ Si selecciona “No se realiza” el valor real es 0%.
- ✓ Si selecciona “Casi se realiza” el valor real es 25%.
- ✓ Si selecciona “Se realiza a medias” el valor real es 50%.
- ✓ Si selecciona “Se realiza casi completo” el valor real es 75%.
- ✓ Si selecciona “Se realiza completo” el valor real es 100%.

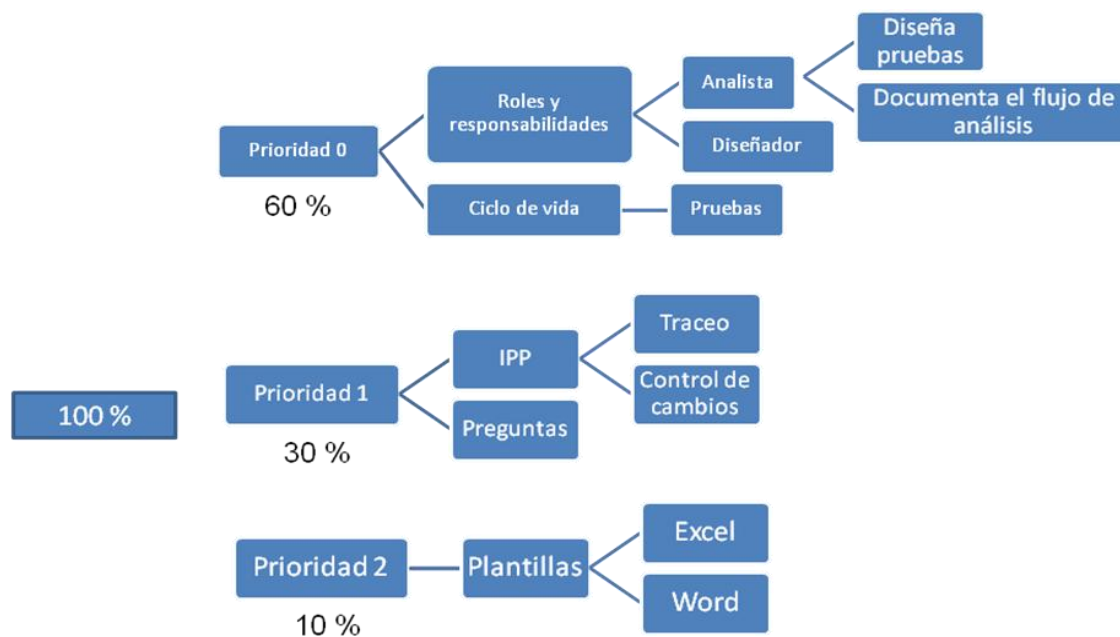
De acuerdo a lo que el evaluador seleccione, el algoritmo calcula lo que representa su valor real de su valor máximo. Cuando calcule todos los valores reales de cada variable se suman y da el porcentaje de acercamiento en que se encuentra realmente el proyecto al programa de mejora.

A continuación se describe un ejemplo del algoritmo para una mejor comprensión, para ello se basa su explicación en la figura 3. La suma de todos los valores máximos de las variables representa el 100%. La sumatoria de todos los valores máximos de las variables de prioridad 0 representa un 60% de este 100%. La sumatoria de todos los valores máximos de las variables de prioridad 1 representa un 30% de este 100%. Y la sumatoria de todos los valores máximos de las variables de prioridad 2 representa un 10 % de este 100%. Siguiendo la figura, prioridad 0 tienen 2 variables: 1ra Rol y responsabilidades y 2da Ciclo de vida.

A su vez Rol y responsabilidades tiene Analista y Diseñador cada una con 15% de valores máximos porque se divide el valor máximo de Rol y Responsabilidades entre la cantidad de variables que lo comprenden es decir  $30/2$

Lo mismo ocurre con Diseña pruebas y Documenta el flujo de análisis cada una con un 7.5% como valores máximos, al dividirse el 15% del Analista entre las 2 variables que lo comprenden. Y así ocurre con todas las demás prioridades.

Si el evaluador selecciona en Diseña pruebas que “Se realiza completa” quiere decir que esta variable se cumple en un 100% es decir en 7.5% que es su valor máximo. Lo mismo ocurre con Documenta el flujo de análisis. El valor real del analista es la suma de los valores reales de Diseña pruebas y Documenta el flujo de análisis. El valor real de Roles y responsabilidades es la suma de los valores reales del Analista y del Diseñador. El valor real de Prioridad 0 es la suma de los valores reales del Ciclo de vida y Rol y responsabilidades. El valor real del proyecto es la suma de los valores reales de Prioridad 0, Prioridad 1 y Prioridad 2. Esto permite cuantificar realmente el estado del proyecto.



**Fig. 3 Ejemplo de variables para calcular el algoritmo.**

El algoritmo a utilizar consiste en calcular el porcentaje que representan los valores reales de cada variable de su valor máximo, y da como resultado la suma de los valores reales, la cual representa el grado de acercamiento del proyecto evaluado al programa de mejora.

### 2.3 Roles de Usuarios

Los roles que interactúan con las funcionalidades de la aplicación son administrador y evaluador. A continuación se relacionan los roles creados que utilizarán en la aplicación:

**Tabla 2.1: Descripción de los Actores.**

Actores	Justificación
Administrador	Persona que tiene acceso a todas las funcionalidades de la aplicación
Evaluador	Persona que tiene acceso a algunas funcionalidades de la aplicación.

Estos roles o actores del sistema van a interactuar con todas las funcionalidades de la aplicación. El Evaluador interactúa con las funcionalidades: Evaluar proyecto, Guardar proyecto, Listar proyecto, Generar reporte, Eliminar proyecto, Cargar proyecto y Adicionar variable. Y el administrador que también es un Evaluador se relaciona con las funcionalidades del Evaluador y: Modifica variable (Adicionar o/y Eliminar variable), Autenticar y Cambiar Contraseña.

A continuación se encuentra el mapa conceptual de roles y funcionalidades.

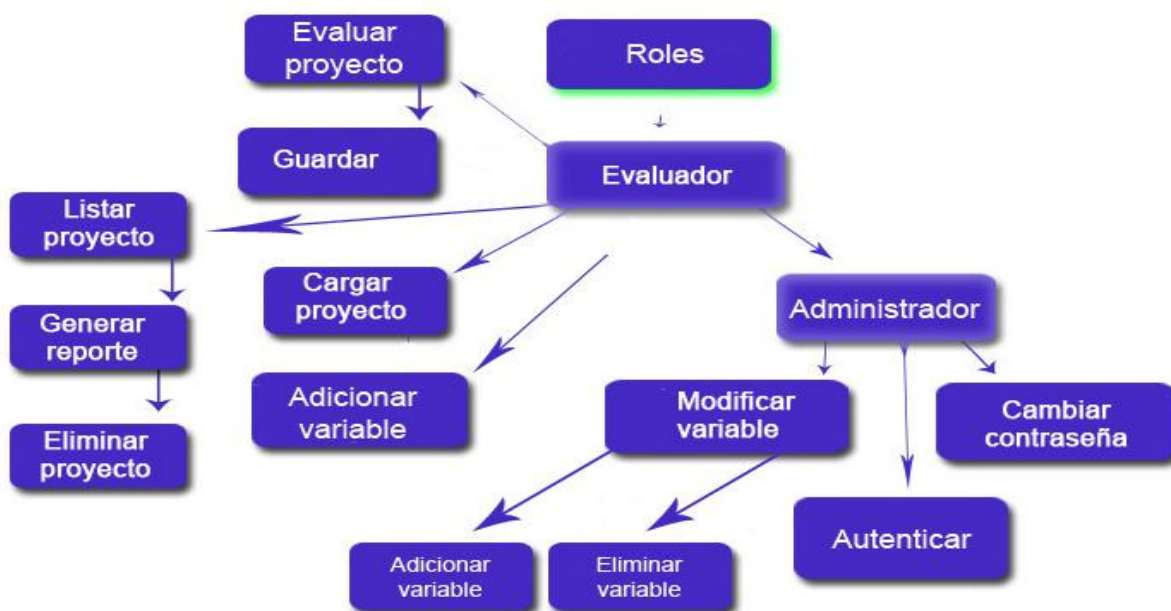


Fig. 4 Mapa conceptual “Roles y Funcionalidades”.

## 2.4 Descripción del proceso de construcción

A continuación se presenta el mapa conceptual Diagrama de Procesos donde se describe el lenguaje de programación que se va a utilizar, los recursos que se utilizan en el proceso, la metodología que se utiliza en el proceso y los roles de la metodología que van a intervenir en el proceso.

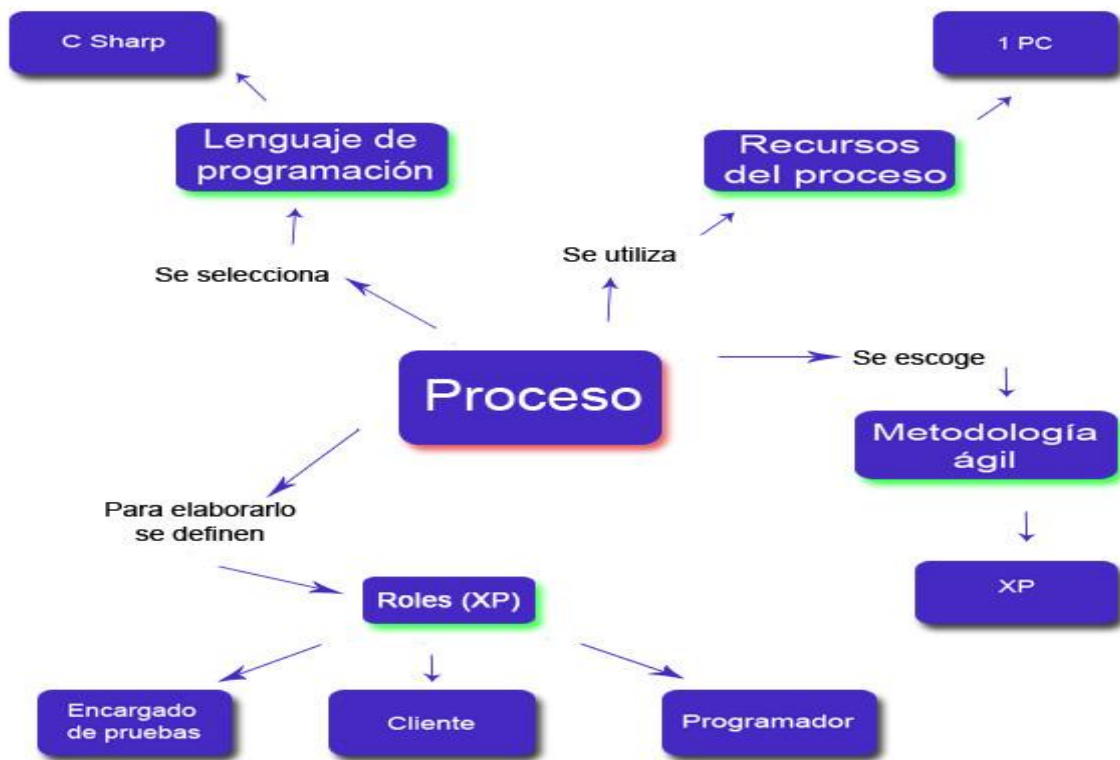


Fig. 5 Mapa conceptual “Diagrama de proceso”.

Para el desarrollo de la aplicación se utilizan los recursos del proceso siguientes (ver figura 5):

- ✓ 1 PC.
- ✓ Como lenguaje de programación C Sharp.
- ✓ XP como metodología ágil.
- ✓ Roles (evaluador o administrador).

Los roles que se utilizan de XP para el desarrollo de la aplicación son los siguientes:

- **Programador (Programmer)**
  - Responsable de decisiones técnicas.



- Responsable de construir el sistema.
  - Sin distinción entre analistas, diseñadores o codificadores.
  - En XP, los programadores diseñan, programan y realizan las pruebas.
- ***Cliente (Customer)***
    - Es parte del equipo.
    - Determina qué construir y cuándo.
- ***Encargado de Pruebas (Tester)***
    - Ayuda al cliente con las pruebas funcionales.
    - Se asegura de que las pruebas funcionales se superan.

La metodología XP define 4 fases fundamentales: Exploración, Planificación, Implementación y Pruebas. En este capítulo se hará alusión a las dos primeras, donde quedarán plasmados los requerimientos del sistema mediante las historias de usuario que describirán en detalle las características de la aplicación definidas previamente. Se definirá además el tiempo total dedicado al proyecto con el objetivo de que la entrega de este se realice sin retrasos en la fecha determinada por el cliente.

## **2.5 Fase de exploración**

La metodología de desarrollo Extreme Programming comienza con la fase de exploración. Durante esta se realiza el proceso de identificación de las historias de usuario, como la familiarización de los equipos de trabajo con las tecnologías y herramientas seleccionadas para la construcción del proyecto.

### **2.5.1 Historias de Usuario**

Uno de los artefactos más importantes que genera la metodología XP son las Historias de Usuario. Estas tienen el mismo propósito que los casos de uso y son escritas por el cliente, como ven ellos las necesidades del sistema, son descripciones cortas y escritas en el lenguaje del usuario sin terminología técnica. Conducen al proceso de creación de los test de aceptación, los cuales sirven para verificar que estas historias se han implementado correctamente. Otra de sus características es que solamente proporcionan los detalles sobre la estimación del riesgo y cuánto tiempo conllevará su implementación. Su

nivel de detalle debe ser el mínimo posible, de manera que permita hacerse una ligera idea de cuánto costará implementar el sistema.

La aplicación cuenta con 10 historias de usuario, cada una respondiendo a las diferentes funcionalidades solicitadas por el cliente y dando una idea al resto del equipo de desarrollo de cómo debe ser su posterior implementación.

A continuación se muestran las historias de usuario definidas para implementar la aplicación:

En esta primera tabla se describe la Historia de Usuario “Autenticar”, la cual interactúa con el administrador, es de prioridad alta al igual que su riesgo en desarrollo, los puntos estimados son de 0.5 semanas, se escriben los nombres de los programadores responsables, se va a implementar en la primera iteración por su prioridad, en la descripción se detalla la misma y en la observación se escriben notas que no son muy importantes pero no se pueden pasar por alto.

**Tabla 2.2: Historia de usuario “Autenticar”.**

<b>Historia de Usuario</b>	
<b>Número: 1</b>	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Autenticar	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 0.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> La aplicación comprueba que el usuario es administrador y le proporciona los privilegios correspondientes.	
<b>Observaciones:</b> En caso de no insertar la contraseña correctamente no accederá a las funcionalidades del administrador.	

En la siguiente Historia de Usuario se describe “Evaluar proyecto”, los usuarios que interactúan con esta Historia son el administrador y el evaluador, de prioridad alta al igual que su riesgo, tiene una semana en

puntos estimados, se implementa en la primera iteración por ser de prioridad alta, se escriben los nombres de los programadores responsables, en la descripción se detalla la tabla y en la misma no es necesario escribirle observaciones.

**Tabla 2.3: Historia de usuario “Evaluar proyecto”.**

<b>Historia de Usuario</b>	
<b>Número: 2</b>	<b>Usuario:</b> Administrador y Evaluador (especialista de calidad, EPG, TWG).
<b>Nombre historia:</b> Evaluar proyecto.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> El usuario selecciona las variables y la posible respuesta para cada una de ellas de acuerdo a las características de su proyecto, cuando haya marcado todas las variables puede evaluar su proyecto.	
<b>Observaciones:</b>	

En la siguiente Historia de Usuario se describe “Adicionar variable”, el usuario que interactúa con esta Historia es el evaluador, de prioridad media al igual que su riesgo, tiene 0.5 semanas en puntos estimados, se implementa en la segunda iteración por ser de prioridad media, se escriben los nombres de los programadores responsables, en la descripción se detalla la tabla y en la observación se escriben notas que no son muy importantes pero no se pueden pasar por alto.

**Tabla 2.4: Historia de usuario “Adicionar variable”.**

<b>Historia de Usuario</b>	
<b>Número: 3</b>	<b>Usuario:</b> Evaluador (especialista de calidad, EPG, TWG)
<b>Nombre historia:</b> Adicionar variable	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 0.5	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> El evaluador puede adicionar variables solamente en las variables de prioridad 0: Etapa de proyecto y Roles y responsabilidades.	
<b>Observaciones:</b> Solamente se puede adicionar en las variables mencionadas anteriormente debido a políticas establecidas por el cliente.	

En la siguiente Historia de Usuario se describe “Cambiar contraseña”, el usuario que interactúa con esta Historia es el administrador, de prioridad baja al igual que su riesgo, tiene 0.5 semanas en puntos estimados, se implementa en la tercera iteración por ser de prioridad baja, se escriben los nombres de los programadores responsables, en la descripción se detalla la tabla y en la observación se escriben notas que no son muy importantes pero no se pueden pasar por alto.

**Tabla 2.5: Historia de usuario “Cambiar contraseña”.**

<b>Historia de Usuario</b>	
<b>Número: 4</b>	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Cambiar contraseña	
<b>Prioridad en negocio:</b> Baja	<b>Riesgo en desarrollo:</b> Baja
<b>Puntos estimados:</b> 0.5	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> La aplicación verifica que la nueva contraseña tenga al menos 7 caracteres.	
<b>Observaciones:</b> El administrador debe introducir la nueva contraseña y esta debe coincidir con la verificación.	

En la siguiente Historia de Usuario se describe “Modificar variable”, el usuario que interactúa con esta Historia es el administrador, de prioridad media al igual que su riesgo, tiene 0.5 semanas en puntos estimados, se implementa en la segunda iteración por ser de prioridad media, se escriben los nombres de los programadores responsables, en la descripción se detalla la tabla y en la observación se escriben notas que no son muy importantes pero no se pueden pasar por alto.

**Tabla 2.6: Historia de usuario “Modificar variable”.**

<b>Historia de Usuario</b>	
<b>Número: 5</b>	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Modificar variable	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 0.5	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> La aplicación comprueba que el usuario sea administrador y le permite modificar (eliminar o adicionar) variables.	
<b>Observaciones:</b> El usuario debe autenticarse como administrador antes de efectuar el cambio.	

En la siguiente Historia de Usuario se describe “Listar proyectos”, los usuarios que interactúan con esta Historia son el administrador y el evaluador, de prioridad media al igual que su riesgo, tiene 0.5 semanas en puntos estimados, se implementa en la segunda iteración por ser de prioridad media, se escriben los nombres de los programadores responsables, en la descripción se detalla la tabla y no tiene observación.

**Tabla 2.7: Historia de usuario “Listar proyectos”.**

<b>Historia de Usuario</b>	
<b>Número: 6</b>	<b>Usuario:</b> Administrador y Evaluador (especialista de calidad, EPG, TWG)
<b>Nombre historia:</b> Listar proyectos	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 0.5	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> La aplicación muestra una lista con todos los proyectos evaluados y el porcentaje que representa cada uno con respecto al proceso de mejora.	
<b>Observaciones:</b>	

En la siguiente Historia de Usuario se describe “Generar reporte”, los usuarios que interactúan con esta Historia son el administrador y el evaluador, de prioridad media al igual que su riesgo, tiene una semana en puntos estimados, se implementa en la segunda iteración por ser de prioridad media, se escriben los nombres de los programadores responsables, en la descripción se detalla la tabla y no tiene observación.

**Tabla 2.8: Historia de usuario “Generar reporte”.**

<b>Historia de Usuario</b>	
<b>Número: 7</b>	<b>Usuario:</b> Administrador y Evaluador (especialista de calidad, EPG, TWG)
<b>Nombre historia:</b> Generar reporte	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> La aplicación guarda un archivo con extensión .pdf con el reporte de los proyectos evaluados (el porcentaje que representan respecto al programa de mejora y las variables cuya evaluación está por debajo del 80%).	
<b>Observaciones:</b>	

En la siguiente Historia de Usuario se describe “Eliminar Proyecto”, los usuarios que interactúa con esta Historia son el administrador y el evaluador, de prioridad media al igual que su riesgo, tiene 0.5 semanas en puntos estimados, se implementa en la segunda iteración por ser de prioridad media, se escriben los nombres de los programadores responsables, en la descripción se detalla la tabla y no tiene observación.



**Tabla 2.9: Historia de usuario “Eliminar Proyecto”.**

<b>Historia de Usuario</b>	
<b>Número: 8</b>	<b>Usuario:</b> Administrador y Evaluador (especialista de calidad, EPG, TWG)
<b>Nombre historia:</b> Eliminar proyecto	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 0.5	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> La aplicación elimina el proyecto seleccionado del listado de proyectos.	
<b>Observaciones:</b>	

En la siguiente Historia de Usuario se describe “Guardar Proyecto”, los usuarios que interactúan con esta Historia son el administrador y el evaluador, de prioridad media al igual que su riesgo, tiene 0.5 semanas en puntos estimados, se implementa en la segunda iteración por ser de prioridad media, se escriben los nombres de los programadores responsables, en la descripción se detalla la tabla y no tiene observación.

**Tabla 2.10: Historia de usuario “Guardar Proyecto”.**

<b>Historia de Usuario</b>	
<b>Número: 8</b>	<b>Usuario:</b> Administrador y Evaluador (especialista de calidad, EPG, TWG)
<b>Nombre historia:</b> Guardar Proyecto	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 0.5	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> La aplicación guarda el proyecto que se está evaluando en ese momento.	
<b>Observaciones:</b>	

En la siguiente Historia de Usuario se describe “Cargar Proyecto”, los usuarios que interactúan con esta Historia son el administrador y el evaluador, de prioridad media al igual que su riesgo, tiene 0.5 semana en puntos estimados, se implementará en la segunda iteración por ser de prioridad media, se escriben los nombres de los programadores responsables, en la descripción se detalla la tabla y no tiene observación.

**Tabla 2.11: Historia de usuario “Cargar Proyecto”.**

<b>Historia de Usuario</b>	
<b>Número: 8</b>	<b>Usuario:</b> Administrador y Evaluador (especialista de calidad, EPG, TWG)
<b>Nombre historia:</b> Cargar Proyecto	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 0.5	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> La aplicación carga el proyecto seleccionado del listado de proyectos guardados.	
<b>Observaciones:</b>	

## 2.6 Fase de Planificación

La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. Es una fase corta, en donde el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que se deberá implementar las historias de usuario, y, asociadas a estas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas.

### 2.6.1 Estimación de esfuerzo por historia de usuario

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar cada historia de usuario. Este se expresa utilizando como medida el punto. Un punto se considera como una semana ideal de trabajo, donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Los resultados obtenidos en esta estimación se exponen en la siguiente tabla:

**Tabla 2.12: Estimación de esfuerzo por historias de usuario.**

<b>Historias de Usuario</b>	<b>Puntos estimados</b>
Autenticar	0.5
Evaluar proyecto.	1
Adicionar variable	0.5
Cambiar contraseña	0.5
Modificar variable	0.5
Listar proyectos	0.5
Generar reporte	1
Eliminar proyecto	0.5
Guardar proyecto	0.5
Cargar proyecto	0.5

## **2.6.2 Plan de Iteraciones**

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación. Así mismo, para cada historia de usuario se establecen las pruebas de aceptación.

Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores. Las pruebas de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección, así como para prever que no vuelvan a ocurrir.

### **2.6.2.1 Iteración 1**

En la primera iteración se implementaron las historias de usuario con mayor prioridad, obteniendo al final una primera versión de prueba y dando a la aplicación las primeras funcionalidades, centrándose en la autenticación y la evaluación de proyectos.

### **2.6.2.2 Iteración 2**

En la segunda iteración se realiza la implementación de las historias de usuario con prioridad media, relacionadas con la modificación de variables, adición de variables (como: etapa de proyecto y, roles y

responsabilidades), listado de proyectos, generar reporte, guardar proyecto, cargar proyecto y eliminar proyecto. De esta forma se obtiene la segunda versión de pruebas de la aplicación.

### 2.6.2.3 Iteración 3

En la tercera iteración se implementan las historias de usuario con prioridad baja, centrándose en cambiar contraseña. Alcanzando la versión 1.0 del producto final.

### 2.6.3 Plan de duración de las iteraciones

El plan que se muestra en la Tabla 2.13 se realiza con el objetivo de reflejar cuales serán las historias de usuario que serán implementadas en cada una de las iteraciones, así como el tiempo destinado a cada una de ellas y el orden en que se implementarán, lo que ayuda a obtener una idea general del tiempo que durará la confección total del sistema.

**Tabla 2.13: Plan de Iteraciones.**

Iteración	Orden de las Historias de Usuario a implementar	Duración de la iteración
Iteración 1	<ol style="list-style-type: none"> <li>1. Autenticar.</li> <li>2. Evaluar proyecto</li> </ol>	1.5 semanas
Iteración 2	<ol style="list-style-type: none"> <li>1. Adicionar variable</li> <li>2. Modificar variable</li> <li>3. Listar proyecto</li> <li>4. Generar reporte</li> <li>5. Eliminar proyecto</li> <li>6. Guardar proyecto</li> <li>7. Cargar proyecto</li> </ol>	4.5 semanas
Iteración 3	<ol style="list-style-type: none"> <li>1. Cambiar contraseña</li> </ol>	0.5 semana

## 2.6 Plan de entregas

De acuerdo al plan que se va exponer a continuación el software culminaría en 7 semanas de implementación es decir las 4 semanas del mes de abril y 3 semanas del mes de mayo. El plan expuesto a continuación se detalla las fechas para las cuales deben ser entregadas cada una de las iteraciones con sus respectivas versiones de prueba.

**Tabla 2.14: Plan de entregas**

No. Historias de Usuarios	Final 1ra Iteración 1era semana de abril	Final 2da Iteración 4ta semana de abril	Final 3ra Iteración 3era semana de mayo
1,2	Versión 0.1		
3,4,5,6,7,9,10		Versión 0.2	
8			Versión 1.0

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Después de definidas las iteraciones, el equipo de desarrollo precisa el orden de las tareas de ingeniería por iteración para cada Historia de Usuario, o sea, se descomponen las Historias de Usuario en tareas de desarrollo; las cuales se asignan a un equipo o una persona responsable de su implementación. Dichas tareas son para el uso de los programadores, pueden escribirse utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente (BECK, 1999).

Al principio de cada iteración se revisa el plan de iteraciones y se modifica de ser necesario (BECK y FOWLER, 2000). Al finalizar cada iteración, cumpliendo con todas las tareas elaboradas, se debe obtener un producto funcional, el cual será probado y mostrado al cliente.

Para el desarrollo de la aplicación se definieron tres iteraciones, en este capítulo se detalla cada iteración así como las tareas de desarrollo y las pruebas de aceptación.

### 3.1 Primera Iteración

En esta iteración se abordan las Historias de Usuario de mayor prioridad (Autenticar y Evaluar Proyecto), el equipo define las tareas de desarrollo con el fin de obtener un producto con las funcionalidades necesarias para mostrar al usuario y obtener criterios de este.

Las Tareas de Ingeniería referente a la primera iteración, han sido resumidas en las tablas a continuación.

Para una mejor comprensión de cómo completar los campos de la tabla de Tareas de Ingeniería, remitirse al Anexo II.

**Tabla 3.1: Historia de Usuario Evaluar proyecto con Estimación y Tiempo Real**

Historia de Usuario	Estimación	Real
Evaluar proyecto.	1	1
<b>Total</b>	1	1

La tabla que se presenta a continuación es la primera Tarea de Ingeniería para la Historia de Usuario Evaluar proyecto.

**Tabla 3.2: Tarea de Ingeniería Nro.1 para la Historia de Usuario Evaluar Proyecto.**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Implementar técnica para evaluar variables	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 20 marzo 2009	<b>Fecha Fin:</b> 23 abril 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Se implementa un algoritmo que muestre el grado de acercamiento que tiene el proyecto que se está evaluando con la solución que está automatizada.	

La tabla que se presenta a continuación es la segunda Tarea de Ingeniería para la Historia de Usuario Evaluar proyecto.



**Tabla 3.3: Tarea de Ingeniería Nro.2 para la Historia de Usuario Evaluar Proyecto.**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Introducir las variables en la aplicación	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 20 febrero 2009	<b>Fecha Fin:</b> 23 febrero 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Las variables (etapa de proyecto y roles y responsabilidades) se introducen en la aplicación de acuerdo a la prioridad dada por el cliente.	

La tabla que se presenta a continuación es la tercera Tarea de Ingeniería para la Historia de Usuario Evaluar proyecto.

**Tabla 3.4: Tarea de Ingeniería Nro.3 para la Historia de Usuario Evaluar Proyecto.**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Implementar técnica para evaluar variables	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 20 marzo 2009	<b>Fecha Fin:</b> 23 abril 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Se implementa un algoritmo que muestre el grado de acercamiento que tiene el proyecto que se está evaluando con la solución que está automatizada.	

La tabla que se presenta a continuación es la cuarta Tarea de Ingeniería para la Historia de Usuario Evaluar proyecto.

**Tabla 3.5: Tarea de Ingeniería Nro.4 para la Historia de Usuario Evaluar Proyecto.**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 3	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Dar resultado de la evaluación	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 20 febrero 2009	<b>Fecha Fin:</b> 23 febrero 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Al finalizar la evaluación de cada proyecto se emite el grado de acercamiento que tiene el proyecto al proceso de mejora.	

**Tabla 3.6: Historia de Usuario Autenticar con su semanas de Estimación y Tiempo Real**

<b>Historia de Usuario</b>	<b>Estimación</b>	<b>Real</b>
Autenticar	0.5	0.5
<b>Total</b>	0.5	0.5

La tabla que se presenta a continuación es la primera Tarea de Ingeniería para la Historia de Usuario Autenticar.

**Tabla 3.7: Tarea de Ingeniería Nro.1 para la Historia de Usuario Autenticar.**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Introducir usuario y contraseña	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 20 febrero 2009	<b>Fecha Fin:</b> 23 febrero 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> El administrador inserta su usuario y contraseña en la aplicación	

La tabla que se presenta a continuación es la segunda Tarea de Ingeniería para la Historia de Usuario Autenticar.

**Tabla 3.8: Tarea de Ingeniería Nro.2 para la Historia de Usuario Autenticar.**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Verificar usuario y contraseña	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 20 febrero 2009	<b>Fecha Fin:</b> 23 febrero 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> La aplicación comprueba que el usuario y la contraseña insertados coincidan con los del fichero, de coincidir se le otorgan los privilegios correspondientes.	

### 3.2 Segunda Iteración

En esta iteración se abordan las Historias de Usuario de prioridad media y se definen las tareas de desarrollo para las mismas.

**Tabla 3.9: Historia de Usuario Adicionar Variable con sus semanas de Estimación y Tiempo Real**

Historia de Usuario	Estimación	Real
Adicionar Variable	0.5	0.5
<b>Total</b>	0.5	0.5

La tabla que se presenta a continuación es la primera Tarea de Ingeniería para la Historia de Usuario Adicionar Variable.

**Tabla 3.10: Tarea de Ingeniería Nro.1 para la Historia de Usuario Adicionar Variable.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Insertar nombre de la nueva variable a adicionar	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 15 marzo 2009	<b>Fecha Fin:</b> 23 marzo 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Se adiciona la nueva variable (etapa de proyecto, roles y responsabilidades) y se guarda en un fichero para utilizar en posibles evaluaciones	

La tabla que se presenta a continuación es la segunda Tarea de Ingeniería para la Historia de Usuario Adicionar Variable.

**Tabla 3.11: Historia de Usuario Modificar Variable con sus semanas de Estimación y Tiempo Real**

Historia de Usuario	Estimación	Real
Modificar Variable	0.5	0.5
<b>Total</b>	0.5	0.5

La tabla que se presenta a continuación es la primera Tarea de Ingeniería para la Historia de Usuario Modificar Variable.

**Tabla 3.12: Tarea de Ingeniería Nro.1 para la Historia de Usuario Modificar Variable.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 5
<b>Nombre Tarea:</b> Seleccionar prioridad	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 24 marzo 2009	<b>Fecha Fin:</b> 30 marzo 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Se selecciona la prioridad de la variable que se desee adicionar	

La tabla que se presenta a continuación es la segunda Tarea de Ingeniería para la Historia de Usuario Modificar Variable.

**Tabla 3.17: Tarea de Ingeniería Nro.2 para la Historia de Usuario Modificar Variable.**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> 5
<b>Nombre Tarea:</b> Seleccionar características de la variable	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 15 marzo 2009	<b>Fecha Fin:</b> 23 marzo 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Se inserta el nombre de la variable y se selecciona el tipo (simple o compuesta)	

La tabla que se presenta a continuación es la tercera Tarea de Ingeniería para la Historia de Usuario Modificar Variable.

**Tabla 3.18: Tarea de Ingeniería Nro.3 para la Historia de Usuario Modificar Variable.**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 3	<b>Número Historia de Usuario:</b> 5
<b>Nombre Tarea:</b> Eliminar	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 15 marzo 2009	<b>Fecha Fin:</b> 23 marzo 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Se selecciona la variable y se elimina.	

La tabla que se presenta a continuación es la cuarta Tarea de Ingeniería para la Historia de Usuario Modificar Variable.

**Tabla 3.19: Tarea de Ingeniería Nro.4 para la Historia de Usuario Modificar Variable.**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 4	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Adicionar	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 15 marzo 2009	<b>Fecha Fin:</b> 23 marzo 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Se adiciona la variable	



**Tabla 3.20: Historia de Usuario Listar proyecto con sus semanas de Estimación y Tiempo Real**

Historia de Usuario	Estimación	Real
Listar proyecto	0.5	0.5
<b>Total</b>	0.5	0.5

La tabla que se presenta a continuación es la primera Tarea de Ingeniería para la Historia de Usuario Listar proyecto.

**Tabla 3.21: Tarea de Ingeniería Nro.1 para la Historia de Usuario Listar proyecto.**

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 6
Nombre Tarea: Listar proyecto	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 15 marzo 2009	Fecha Fin: 23 marzo 2009
Programador Responsable: Elisandra Soria – Arianne Prades	
Descripción: Se carga un listado con todos los proyectos evaluados	

**Tabla 3.22: Historia de Usuario Generar reporte con sus semanas de Estimación y Tiempo Real**

Historia de Usuario	Estimación	Real
Generar reporte	1	1
<b>Total</b>	1	1

La tabla que se presenta a continuación es la primera Tarea de Ingeniería para la Historia de Usuario Generar reporte.

**Tabla 3.23: Tarea de Ingeniería Nro.1 para la Historia de Usuario Generar reporte.**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 7
<b>Nombre Tarea:</b> Generar reporte	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 15 marzo 2009	<b>Fecha Fin:</b> 23 marzo 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Se guarda en un archivo con extensión .pdf la evaluación del proyecto y las variables evaluadas en menos de un 80%	

**Tabla 3.24: Historia de Usuario Eliminar proyecto con sus semanas de Estimación y Tiempo Real**

<b>Historia de Usuario</b>	<b>Estimación</b>	<b>Real</b>
Eliminar proyecto	0.5	0.5
<b>Total</b>	0.5	0.5

La tabla que se presenta a continuación es la primera Tarea de Ingeniería para la Historia de Usuario Eliminar proyecto.

**Tabla 3.25: Tarea de Ingeniería Nro.1 para la Historia de Usuario Eliminar proyecto.**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 8
<b>Nombre Tarea:</b> Eliminar proyecto	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 15 marzo 2009	<b>Fecha Fin:</b> 18 marzo 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Se elimina el proyecto seleccionado del listado de proyectos evaluados	

**Tabla 3.26: Historia de Usuario Guardar proyecto con sus semanas de Estimación y Tiempo Real**

<b>Historia de Usuario</b>	<b>Estimación</b>	<b>Real</b>
Guardar proyecto	0.5	0.5
<b>Total</b>	0.5	0.5

La tabla que se presenta a continuación es la primera Tarea de Ingeniería para la Historia de Usuario Guardar proyecto.

**Tabla 3.27: Tarea de Ingeniería Nro.1 para la Historia de Usuario Guardar proyecto.**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 9
<b>Nombre Tarea:</b> Guardar proyecto	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 19 marzo 2009	<b>Fecha Fin:</b> 23 marzo 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Se guarda el proyecto con sus variables (evaluadas y sin evaluar).	

**Tabla 3.28: Historia de Usuario Cargar proyecto con sus semanas de Estimación y Tiempo Real**

<b>Historia de Usuario</b>	<b>Estimación</b>	<b>Real</b>
Cargar proyecto	0.5	0.5
<b>Total</b>	0.5	0.5

La tabla que se presenta a continuación es la primera Tarea de Ingeniería para la Historia de Usuario Cargar proyecto.

**Tabla 3.29: Tarea de Ingeniería Nro.1 para la Historia de Usuario Cargar proyecto.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 9
<b>Nombre Tarea:</b> Cargar proyecto	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 23 marzo 2009	<b>Fecha Fin:</b> 30 marzo 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> Se carga el proyecto del listado de proyectos guardados.	

### 3.3 Tercera Iteración

En esta iteración se aborda la Historia de Usuario de prioridad baja (ver Tabla 40-41). Concluida esta iteración, se consta de un producto listo para su puesta en funcionamiento.

**Tabla 3.30: Historia de Usuario Cambiar contraseña con sus semanas de Estimación y Tiempo Real**

Historia de Usuario	Estimación	Real
Cambiar contraseña	0.5	0.5
<b>Total</b>	0.5	0.5

La tabla que se presenta a continuación es la primera Tarea de Ingeniería para la Historia de Usuario Cambiar contraseña.

**Tabla 3.31: Tarea de Ingeniería Nro.1 para la Historia de Usuario Cambiar contraseña.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 4
<b>Nombre Tarea:</b> Verificar datos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 1ro de abril 2009	<b>Fecha Fin:</b> 4 de abril 2009
<b>Programador Responsable:</b> Elisandra Soria – Arianne Prades	
<b>Descripción:</b> El administrador debe insertar la nueva contraseña y la misma debe coincidir con la verificación.	

### 3.4 Pruebas

Entre las prácticas de XP está la de seguir un desarrollo guiado por pruebas, posibilitando darle al cliente una idea de las verdaderas funcionalidades que tiene el producto. Mediante esta filosofía se reduce el número de errores no detectados, así como el tiempo entre la introducción de este en el sistema y su detección (Crispin y House, 2002). Realizándole al producto pruebas constantemente al final de cada iteración se eleva la calidad del mismo.

XP divide las pruebas en tres grupos:

**Pruebas unitarias:** desarrolladas por los programadores y encargadas de verificar el código de forma automática.

**Pruebas de aceptación:** destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente. Por cada iteración se traducen las Historias de Usuarios en pruebas de aceptación, con el objetivo de demostrar al cliente que el producto cumple con los requerimientos necesarios. Por cada Historia de Usuario se pueden realizar todas las pruebas que se estimen convenientes.

**Pruebas de Integración:**

A continuación se mostrarán las pruebas de aceptación realizadas al sistema. Detalles de cómo completar los campos de la tabla de Caso de Prueba de Aceptación son mostrados en el Anexo III.

En la primera 1ra se aceptación se describe el caso de prueba para la 1ra Historia de Usuario.

**Tabla 3.32: Caso de Prueba de Aceptación HU1P1.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU1P1	<b>Número Historia de Usuario:</b> 1
<b>Nombre:</b> Evaluar el proyecto y verificar si el algoritmo cumple con lo establecido	
<b>Descripción de la Prueba:</b> Se selecciona la respuesta más acertada de acuerdo al cumplimiento que tiene la variable en ese proyecto.	
<b>Condiciones de ejecución:</b> Se debe seleccionar al menos una variable	
<b>Entrada/Pasos Ejecución:</b> El evaluador debe evaluar todas las variables que utiliza en su proyecto y evaluar el proyecto	
<b>Resultado Esperado:</b> Obtener grado de acercamiento del proyecto al proceso de mejora.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

En la 2da prueba se aceptación se describe el caso de prueba para la 2da Historia de Usuario.

**Tabla 3.33: Caso de Prueba de Aceptación HU2P1.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU2P1	<b>Número Historia de Usuario:</b> 2
<b>Nombre:</b> Escribir y verificar si coincide la contraseña guardada en el fichero	
<b>Descripción de la Prueba:</b> La contraseña escrita es encriptada en MD5 y verificada con el texto guardado ya encriptado.	
<b>Condiciones de ejecución:</b> La contraseña entrada por el administrador debe coincidir con la contraseña guardada	
<b>Entrada/Pasos Ejecución:</b> El administrador debe introducir su contraseña correctamente	

<b>Resultado Esperado:</b> Obtener el acceso a los privilegios del administrador
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.

En la 3ra prueba se aceptación se describe el caso de prueba para la 3ra Historia de Usuario.

**Tabla 3.34: Caso de Prueba de Aceptación HU3P1.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU3P1	<b>Número Historia de Usuario:</b> 3
<b>Nombre:</b> Insertar variable	
<b>Descripción de la Prueba:</b> Se verifica si existen variables con el mismo nombre	
<b>Condiciones de ejecución:</b> Se adiciona la variable en etapa de proyecto y roles y responsabilidades	
<b>Entrada/Pasos Ejecución:</b> Se verifica si la variable adicionada por el evaluador está en el listado del proyecto del cliente	
<b>Resultado Esperado:</b> Éxito al adicionar la variable	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

En la 4ta prueba se aceptación se describe el caso de prueba para la 4ta Historia de Usuario.

**Tabla 3.35: Caso de Prueba de Aceptación HU4P1.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU4P1	<b>Número Historia de Usuario:</b> 4
<b>Nombre:</b> Verificar que se ha cambiado la contraseña	
<b>Descripción de la Prueba:</b> La contraseña es insertada 2 veces a modo de verificación, encriptada en MD5 y guardada en el fichero	
<b>Condiciones de ejecución:</b> La nueva contraseña debe tener al menos 7 caracteres	
<b>Entrada/Pasos Ejecución:</b> Se verifica si la contraseña se guarda correctamente y da acceso a los privilegios del administrador	
<b>Resultado Esperado:</b> Cambio de contraseña correcta.	



**Evaluación de la Prueba:** Prueba satisfactoria.

En la 5ta prueba se aceptación se describe el caso de prueba para la 5ta Historia de Usuario.

**Tabla 3.36: Caso de Prueba de Aceptación HU5P1.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU5P1	<b>Número Historia de Usuario:</b> 5
<b>Nombre:</b> Insertar variable	
<b>Descripción de la Prueba:</b> Se verifica si existen variables con el mismo nombre	
<b>Condiciones de ejecución:</b> Se adiciona la variable con sus características	
<b>Entrada/Pasos Ejecución:</b> Se verifica si la variable adicionada por el administrador está en el listado de la propuesta de solución para los proyectos.	
<b>Resultado Esperado:</b> Éxito al adicionar la variable	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

En la 6ta prueba se aceptación se describe el caso de prueba para la 5ta Historia de Usuario.

**Tabla 3.37: Caso de Prueba de Aceptación HU5P2.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU5P2	<b>Número Historia de Usuario:</b> 5
<b>Nombre:</b> Eliminar variable	
<b>Descripción de la Prueba:</b> Se comprueba si la variable seleccionada es eliminada	
<b>Condiciones de ejecución:</b> La variable debe estar en el listado	
<b>Entrada/Pasos Ejecución:</b> Se confirma que una vez eliminada la variable, esta no aparezca en el listado de variables de la propuesta de solución.	
<b>Resultado Esperado:</b> Variable eliminada.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

En la 7ma prueba se aceptación se describe el caso de prueba para la 6ta Historia de Usuario.

**Tabla 3.38: Caso de Prueba de Aceptación HU6P1.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU6P1	<b>Número Historia de Usuario:</b> 6
<b>Nombre:</b> Mostrar listado de proyectos evaluados	
<b>Descripción de la Prueba:</b> Se comprueba si el proyecto evaluado es guardado en el fichero	
<b>Condiciones de ejecución:</b> El proyecto debe estar evaluado para que aparezca en el listado	
<b>Entrada/Pasos Ejecución:</b> Se lista el proyecto con sus datos	
<b>Resultado Esperado:</b> Proyecto listado.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

En la 8va prueba se aceptación se describe el caso de prueba para la 7ma Historia de Usuario.

**Tabla 3.39: Caso de Prueba de Aceptación HU7P1.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU7P1	<b>Número Historia de Usuario:</b> 7
<b>Nombre:</b> Generar resultados de los proyectos evaluados	
<b>Descripción de la Prueba:</b> Se genera un archivo .pdf donde se encuentran las variables con menos de un 80% de cumplimiento	
<b>Condiciones de ejecución:</b> El proyecto debe estar evaluado.	
<b>Entrada/Pasos Ejecución:</b> Seleccionar el proyecto del listado de proyectos y generar el reporte	
<b>Resultado Esperado:</b> Reporte guardado correctamente	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

En la 90 prueba se aceptación se describe el caso de prueba para la 8va Historia de Usuario.

**Tabla 3.40: Caso de Prueba de Aceptación HU8P1.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU8P1	<b>Número Historia de Usuario:</b> 8
<b>Nombre:</b> Eliminar proyecto del listado de proyectos evaluados	
<b>Descripción de la Prueba:</b> Se comprueba si el proyecto seleccionado es eliminado	
<b>Condiciones de ejecución:</b> El proyecto debe estar en el listado de proyectos evaluados	
<b>Entrada/Pasos Ejecución:</b> Se confirma que una vez eliminado el proyecto, este no aparezca en el listado de proyectos evaluados.	
<b>Resultado Esperado:</b> Proyecto eliminado.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

En la 10ma prueba se aceptación se describe el caso de prueba para la 9na Historia de Usuario.

**Tabla 3.41: Caso de Prueba de Aceptación HU9P1.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU9P1	<b>Número Historia de Usuario:</b> 9
<b>Nombre:</b> Guardar proyecto que se está evaluando.	
<b>Descripción de la Prueba:</b> Se comprueba si el proyecto seleccionado es guardado.	
<b>Condiciones de ejecución:</b> El proyecto debe estar en el listado de proyectos guardados.	
<b>Entrada/Pasos Ejecución:</b> Se confirma que una vez guardado el proyecto, este aparezca en el listado de proyectos guardados.	
<b>Resultado Esperado:</b> Proyecto guardado.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

En la 11na prueba se aceptación se describe el caso de prueba para la 10ma Historia de Usuario.

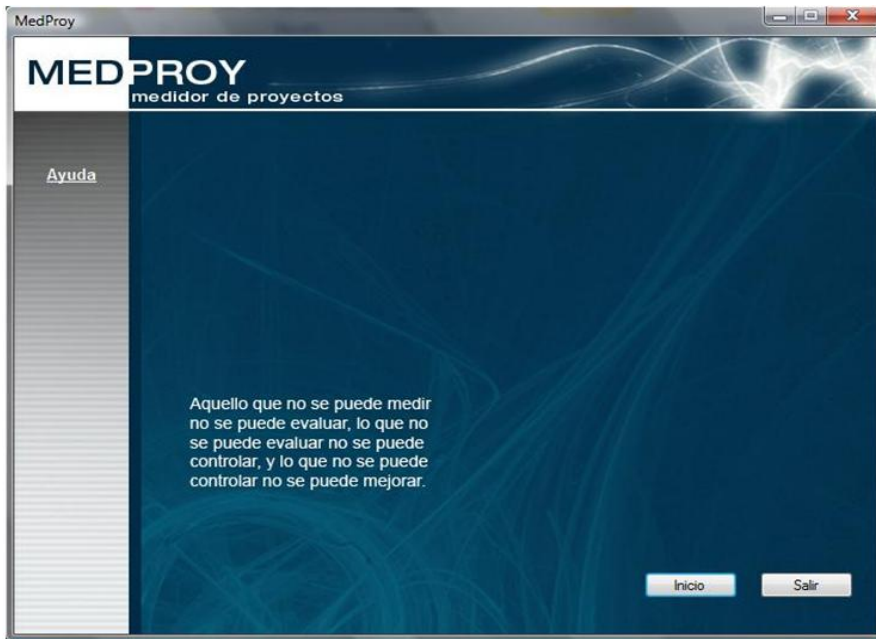
**Tabla 3.42: Caso de Prueba de Aceptación HU10P1.**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU10P1	<b>Número Historia de Usuario:</b> 10
<b>Nombre:</b> Cargar proyecto del listado de proyectos guardados.	
<b>Descripción de la Prueba:</b> Se comprueba si el proyecto seleccionado es cargado.	
<b>Condiciones de ejecución:</b> El proyecto debe estar en el listado de proyectos guardados.	
<b>Entrada/Pasos Ejecución:</b> Se confirma que una vez cargado el proyecto, este se encuentre en las mismas condiciones con que se guardo.	
<b>Resultado Esperado:</b> Proyecto cargado.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

### **3.5 Diseño de la Aplicación**

Para el diseño de la aplicación se tomaron en cuenta los colores predeterminados de Windows. El diseño es bastante serio y sencillo ya que la aplicación lo requiere. A continuación se presentan los formularios de la aplicación:

Este primer formulario es la presentación de la aplicación, con la ayuda presente, el botón de Inicio y Salir.



**Fig. 6** Formulario “Presentación de la aplicación”.

El siguiente formulario presenta todas las funcionalidades de la aplicación.



**Fig. 7** Formulario “Funcionalidades de la aplicación”.

El siguiente formulario presenta la Evaluación de los proyectos.

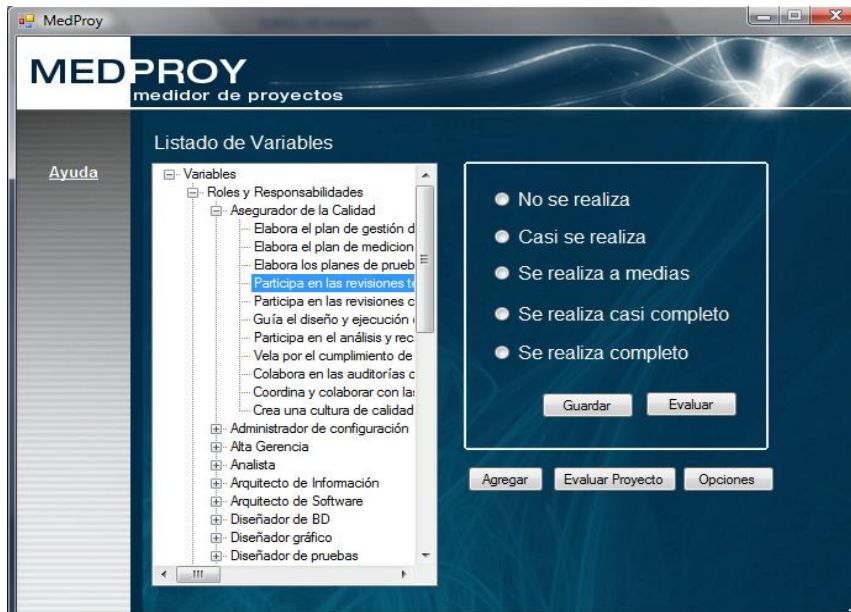


Fig. 8 Formulario “Evaluación de Proyectos”.

El siguiente formulario presenta la forma en que se adicionan y se eliminan las variables según su prioridad.



Fig. 9 Formulario “Adicionar y Eliminar Variables”.

## CONCLUSIONES

Mediante el estudio realizado y la aplicación realizada se puede concluir:

- ✓ Se implementó una aplicación que ayude a diagnosticar el grado de acercamiento de los proyectos productivos al programa de mejora continua en la UCI, permitiendo asegurar la calidad de los proyectos y establecer como deberán funcionar correctamente.
- ✓ La evaluación de variables posibilita conocer el estado real del proyecto evaluado.
- ✓ La propuesta de solución fue validada satisfactoriamente por el cliente cumpliendo con el objetivo trazado.
- ✓ La metodología XP fue usada para la elaboración de un proyecto pequeño brindando resultados satisfactorios.



## RECOMENDACIONES

Teniendo en cuenta la importancia de la aplicación se recomienda:

- ✓ Aplicar la propuesta de solución a todos los proyectos productivos de la UCI.
- ✓ Realizar actualizaciones a la aplicación para que soporte otros niveles de CMMI.
- ✓ Proponer otra versión creando un rol que se llame invitado para que cada proyecto sea el que se haga la evaluación.

## BIBLIOGRAFÍA

1. Almeraz, E. (2006) Lecciones aprendidas y estrategias de Implantación de CMMI® en México.
2. **Caballero Rúa, Edgar Henry.** *Mejora de la calidad del software en el entorno de microempresas TI.* José Antonio Calvo Manzano Villalón. Universidad Politécnica de Madrid Facultad de Informática, 2006-2007.
3. **Canlı yayımlar, Aktif seyircilerle sohbet.** *Lenguajes de programación.* Disponible en: <http://www.lenguajes-de-programacion.com/programacion-orientada-a-objetos.shtml>.
4. **CANÓS, José H. y LETELIER, Patricio y PENADÉS, M<sup>a</sup> Carmen (2008).** *Métodologías Ágiles en el Desarrollo de Software.* [fecha de consulta: 10 junio 2008]. Disponible en: <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>.
5. **Castro, Franklin.** Que Ciencia. Disponible en <http://www.queciencia.com/2007/10/09/la-importancia-de-asegurar-la-calidad-del-software/>.
6. Dávila, A. (2005) Determinación de los Requerimientos de Calidad del Producto Software Basados en Normas Internacionales.
7. **Eduardo Medina, Antonio J. Solís.** 2004. La necesidad de un sistema de la calidad para prevenir y controlar los problemas del software. Ingenia. Parque Tecnológico de Andalucía. Noviembre de 2004. Disponible en: <http://www.arqui.com/>.
8. Gabriel Fernández, J. T. (2008) Modelo de Mejora de Proceso de Software.
9. Garro, I. (2005) Calidad de Software: Modelos y estándares.
10. Guerra, R. T. (2005) CALIDAD EN EL DESARROLLO DE SOFTWARE.  
<http://www.willydev.net/InsiteCreation/v1.0/descargas/masyxp.pdf>.
11. **XP Extreme Programming: A gentle introduction.** 17 de Febrero 2006. Disponible en: <http://www.extremeprogramming.org/>.
12. **Eguiluz Perez, Javier.** 2008. Disponible en: <http://www.librosweb.es/javascript/index.html>.
13. **INFOCALIDAD.** 2004. Disponible en: <http://www.infocalidad.net/principal/index.asp>.
14. **Iván Nieto Pérez** *El código.* 2007. Disponible en <http://www.elcodigo.net/tutoriales/javascript/javascript1.html#punto1>.
15. **LETELIER, Patricio y PENADÉS, Ma. Carmen. (2008).** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* [en línea]. [fecha consultada: 5 junio 2008]. Disponible en:
16. M., C. P. (2008). Análisis descriptivo del proceso de implementación del nivel 2 del modelo CMMI en una empresa regional de desarrollo de software.

17. **Maestros del Web.** Noviembre 2, 2007. Disponible en: <http://www.maestrosdelweb.com>.
18. Mejora Continua en el Desarrollo del Software Disponible en: <http://www.monografias.com/trabajos32/mejora-continua-software/mejora-continua-software.shtml>.
19. **Microsoft Corporation.** *Inicio de .NET Framework.* 2009 Disponible en: <http://msdn.microsoft.com/es-es/library/aa139615.aspx>.
20. **MIGUEL, B.** *El proyecto informático de construcción de software.* 2004.
21. Natalia Juristo, A. M. M., Sira Vegas (2005). TÉCNICAS DE EVALUACIÓN DE SOFTWARE.
22. Paz, J. D. (2005) Desarrollo en comunidad con eXtreme Programming.
23. Penadés, P. L. y. M. C. (2005) Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
24. Peralta, M. D., E.; Britos, P. y García Martínez, R. (2006) Evaluación asistida de CMMI-SW.
25. Peralta, M. L. (2006) ASISTENTE PARA LA EVALUACIÓN DE CMMI-SW.
26. Pilar Gómez Gil, P. (2007) Un Camino Hacia el Éxito Mundial en el Desarrollo del Software Mexicano.
27. **PROGRAMACIÓN, L. D.** *Programación Java,* 2006. Disponible en: <http://www.lenguajes-de-programacion.com/programacion-java.shtml>.
28. **ROMÁN, C. A.** *La Tecnología Java,* 2007. Disponible en: [http://profesores.fi-b.unam.mx/carlos/java/java\\_basico1\\_1.html](http://profesores.fi-b.unam.mx/carlos/java/java_basico1_1.html).
29. **Scalone, Fernanda.** Estudio Comparativo de los Modelos y Estándares de Calidad del Software. Buenos Aires: s.n., Junio 2006. Tesis Doctoral.
30. *Software Lifecycle Optimization.* 2007 Disponible en: <http://www.als-es.com/home.php?location=recursos/articulos/mejora-procesos-cmmi>.

## ANEXOS

### **Anexo 1:** Guía de la entrevista

Se realiza con el propósito de conocer:

- Funcionalidades del software.
- Definición de variables e indicadores.
- Prioridades a considerar relacionadas con las variables.
- Distintas maneras de preguntar como evaluara el software cada variable.
- Conocer los roles que van a interactuar con el sistema.
- Si el diseño es aceptable para la aplicación a desarrollar.

### Entrevista

Preguntas de las entrevistas:

¿Cuáles son los objetivos de la aplicación a elaborar?

¿Qué funcionalidades debe tener el software?

¿Qué se define como variable e indicador?

¿Están distribuidas las variables de la misma prioridad con el mismo peso?

¿Cuáles son las posibles respuestas que dará el software por cada variable existente en el mismo?

¿Qué usuarios van a interactuar con el sistema?

¿Es aceptable la propuesta del diseño para la aplicación a desarrollar?

**Anexo 2.** Plantilla para especificación de Historias de Usuario.

Historia de Usuario	
Número:	Nombre Historia de Usuario:
Usuario:	
Prioridad en Negocio:	Riesgo en Desarrollo:
Puntos Estimados:	Iteración Asignada:
Descripción:	
Observaciones:	

**Número:** Número de la historia.

**Usuario:** El tipo de usuario que la ejecuta.

**Nombre Historia de Usuario:** Nombre que identifica la historia.

**Prioridad en negocio:** Prioridad que tiene la historia para la entidad. (Según cliente)

**Riesgo en desarrollo:** Grado de dificultad de implementación estimado. (Según programadores)

**Puntos estimados:** Tiempo en semanas que se le asignará. (Estimado)

**Iteración asignada:** En qué iteración se desarrollará. (Según su importancia)

**Descripción:** Breve descripción del proceso que define la historia.

**Observaciones:** Alguna acotación importante de señalar acerca de la historia.

### Anexo 3. Plantilla para especificación de Tareas de ingeniería.

Tarea de Ingeniería	
Número Tarea:	Número Historia de Usuario:
Nombre Tarea:	
Tipo de Tarea:	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable:	
Descripción:	

**Número Tarea:** Número de la tarea de ingeniería.

**Número Historia de Usuario:** Número que identifica la historia a la que pertenece la tarea.

**Nombre Tarea:** Nombre que identifica a la tarea.

**Tipo de Tarea:** Desarrollo / Corrección / Mejora / Otra (especificar).

**Puntos estimados:** Tiempo en semanas que se le asignará. (Estimado).

**Fecha Inicio:** Fecha de inicio de elaboración de la tarea.

**Fecha Fin:** Fecha donde concluye la elaboración de la tarea.

**Programador responsable:** Programador responsable de su implementación.

**Descripción:** Breve descripción del proceso que define la tarea.

**Anexo 4.** Plantilla para especificación para los Casos de Prueba de Aceptación.

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Número Historia de Usuario:
Nombre:	
Descripción de la Prueba:	
Condiciones de Ejecución	
Entrada / Pasos de ejecución:	
Resultado Esperado:	
Evaluación de la Prueba:	

**Código Caso de Prueba:** Código del Caso de Prueba de Aceptación.

**Número Historia de Usuario:** Número que identifica la historia a la que se le realizará la prueba.

**Nombre:** Nombre que define el Caso de Prueba de Aceptación.

**Descripción de la Prueba:** Breve descripción de los pasos a seguir para realizar la prueba.

**Condiciones de Ejecución:** Condiciones iniciales con las que debe cumplir el sistema o parte del sistema a desarrollar, para poder realizar la prueba.

**Entrada / Pasos de ejecución:** Pasos a seguir para lograr la correcta ejecución de la prueba.

**Resultados Esperados:** Descripción de los resultados que se esperan obtener.

**Evaluación de la Prueba:** Definir si fueron logrado o no los resultados esperados.



## GLOSARIO DE TERMINOS

**Base de Datos:** Conjunto organizado e integrado de datos almacenados en computadora, con el fin de facilitar su uso para aplicaciones con múltiples finalidades, de manera que sea posible acceder a la información de forma fácil y rápida.

**CPU:** Unidad central de procesamiento.

**sesgo:** es un error que aparece en los resultados de un estudio debido a factores que dependen de la recolección, análisis, interpretación, publicación o revisión de los datos que pueden conducir a conclusiones que son sistemáticamente diferentes de la verdad o incorrectas acerca de los objetivos de una investigación. Este error puede ser sistemático o no, y es diferente al error aleatorio.

**Frameworks:** en el desarrollo de software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**MSG:**

**EGP:** engineer process group.

**TWG:** time work group.

**QA:** aseguramiento de la calidad o, en inglés **Quality Assurance**.

**clases gráficas awt:** AWT significa Abstract Windowing Toolkit. Permite crear entornos gráficos de ventanas a la manera de Windows.

**clases gráficas swing:** es actualmente el paquete más utilizado para construir interfaces gráficas.

**JDBC:** *Java Database Connectivity*, más conocida por sus siglas **JDBC**, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java.

**librerías dinámicas:** es simplemente un trozo de código que contiene recursos preconstruidos; recursos que pueden ser utilizados en un ejecutable.

**IDEs:** entorno de desarrollo integrado o, en inglés, *Integrated Development Environment ('IDE')*, es un programa compuesto por un conjunto de herramientas para un programador. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.