



UNIVERSIDAD NACIONAL DE INGENIERIA
DIRECCIÓN DE ESTUDIOS DE POSGRADO
MAESTRIA INFORMÁTICA EMPRESARIAL

Tesis para la obtención del grado de
Master
Informática Empresarial

*“Propuesta de implementación de la metodología de Programación Extrema
en la gestión de los sistemas de información en la Unidad de Sistemas
Tributarios de la Dirección General de Ingresos.”*

Elaborado por:

✓ Ing. Estefan Antonio Menocal Zamora.

Tutor de tesis:

✓ Msc. Walger Herrera Treminio.

Managua, Nicaragua, mayo de 2019.

Dedicatoria

A Dios, por permitirme cumplir este logro, por darme las energías, salud necesaria para culminar todo lo que me he propuesto, a mis padres por estar a mi lado y darme su apoyo incondicional en todo lo que necesite y a Yerling Ninoska Altamirano Menocal, mi novia que ha estado siempre dándome motivación, aliento en los momentos difíciles y estar siempre disponible en lo que necesite.

Agradecimiento

A Dios, por darme la salud necesaria para permitirme cumplir este logro, a mis padres por estar a mi lado y darme su apoyo incondicional en todo lo que necesite y a Yerling Ninoska Altamirano Menocal, mi novia que ha estado siempre dándome motivación, aliento en los momentos difíciles y estar siempre disponible en lo que necesite.

Managua, martes 25 de febrero del 2019

Constancia

A quien concierne:

Por este medio, hago constar que he realizado la revisión de la tesis: **“Propuesta de implementación de metodología de programación extrema en la gestión de sistemas de información en la unidad de sistemas tributarios de la DGI.”** en la cual participé como tutor, considero que dicho trabajo cumple con los requerimientos necesarios para su defensa.

Autor del trabajo:

- Ing. Estefan Antonio Menocal Zamora

Sin nada más a que referirme, me despido, deseándole éxitos en sus funciones.

Atte,



Msc. Ing. Walger Herrera Treminio
Docente DEPEC

Resumen

El presente trabajo de tesis tiene como contenido analizar el procedimiento de desarrollo de sistemas en la División de Informática y Sistemas de la Dirección General de Ingresos. Se realizó un diagnóstico de la situación actual en cuanto a la forma de trabajo, dando como resultados que el proceso de gestión de desarrollo de sistemas tiene deficiencias.

Para la solución se propone la implementación de la metodología de desarrollo ágil de Programación Extrema (XP), desarrollando el flujo del proceso que la metodología establece en sus cuatro etapas: planeación, diseño, codificación y pruebas.

Se desarrolló la propuesta del nuevo flujo de trabajo que implementa la metodología de desarrollo ágil de programación extrema, aplicando todas sus fases y herramientas que se deben seguir para que la División de Informática y Sistemas tenga un proceso estándar y organizado de desarrollo de sistemas, que le permita eliminar las deficiencias en el proceso, se analice mejor los requerimientos y optimicen el tiempo de desarrollo y dirijan esfuerzos a otros proyectos. Por último se determinaron los costos de inversión en que se incurrirían si se implementara la metodología.

Se determinó que la metodología aplica al tipo y naturaleza del proceso que se desarrolla en la División de Informática, ya que esta metodología propone un desarrollo ágil, con flexibilidad en cambios de los requerimientos y elaboración de documentación.

Palabras claves: XP, Programación Extrema, Requerimiento, Historias de Usuario.

Índice

I. Introducción	1
II. Antecedentes	2
III. Planteamiento del problema	3
IV. Justificación.....	4
V. Objetivos	5
Capítulo I: Marco Teórico	6
1. Sistemas de información.....	7
1.1. Tipos de sistemas de información.....	8
1.2. Sistemas de información y procesamiento de datos.....	8
1.3. Procesos de sistemas de información.....	8
2. Metodología de Programación Extrema (XP)	9
2.1. Valores XP.....	10
2.2. Características	10
2.3. Roles De La Metodología XP	11
2.4. Fases de la Metodología XP.....	12
2.4.1. Planeación	12
2.4.2. Diseño	13
2.4.3. Codificación.....	14
2.4.4. Pruebas.....	14
3. Ingeniería Web.....	16
3.1. Proceso de la Ingeniería Web.....	16
3.2. Diferencia con la Ingeniería de Software.....	17
3.3. Tecnología en el desarrollo web.....	18
3.4. Ingeniería web basada en UML (UWE.)	22
3.4.1. Modelos.....	22
Capítulo II: Desarrollo y justificación de la solución propuesta	29
Aspectos organizacionales	30
Visión	30
Misión	30
Estructura Organizacional.....	31

Enfoque sistémico.....	32
Diagrama causa y efecto de la problemática.....	32
Definición de actores.....	33
Estructura de la línea de mando o jerarquía de puesto.....	36
Proceso actual.....	36
Resultados de la Entrevista.....	38
Conclusiones del diagnóstico.....	38
Propuesta de aplicación de la metodología Programación Extrema (XP).....	39
Introducción a la Metodología.....	39
Planeación.....	40
Plan de entrega del proyecto.....	43
Ciclo de vida del sistema.....	44
Diagrama de actividad del proceso propuesto.....	49
Ejemplo de aplicación de la metodología de Programación Extrema.....	50
Capítulo III: Evaluación de costos de la metodología.....	62
Estimación de costos.....	63
Equipos disponibles.....	63
Servicios disponibles.....	63
Gastos en papelería.....	64
Gastos en capacitación.....	65
Costos totales de inversión.....	65
Beneficios.....	66
Conclusiones.....	67
Recomendaciones.....	68
Glosario.....	69
Bibliografía.....	70
Anexos.....	74

Introducción

La Dirección General de Ingresos, es una institución descentralizada con autonomía administrativa y financiera, cuyo objetivo es recaudar los tributos internos con equidad, transparencia y eficiencia, promoviendo la cultura Tributaria y cumplimiento de las Leyes, actos y disposiciones que establecen o regulan ingresos a favor del Estado, aportando recursos para el desarrollo económico y social del país.

En su estructura organizacional está seccionada en tres bloques, primero están las áreas de apoyo a la Dirección Superior, luego están las Divisiones de áreas de apoyo a la administración y por último están las Direcciones de Áreas Sustantivas.

Dentro de las divisiones esta la División de Informática y Sistemas (DIS), esta se compone de tres oficinas: Oficina de Sistemas Informáticos, Oficina de Seguridad Informática y la Oficina de Apoyo Tecnológico.

La Oficina de Sistemas se divide en tres unidades entre ellas, unidad de Control de Calidad (UCC), Unidad de Sistemas Tributarios (UST) y Unidad de Sistemas de Apoyo (USA).

El presente estudio se desarrolla en la Unidad de Sistemas Tributarios, teniendo como problemática principal que la ejecución de proyectos de desarrollo de sistemas se vuelve complejo gestionar, debido a la gran cantidad de proyectos en desarrollo de forma simultánea, muchos con plazos extremadamente cortos, esto hace que algunos proyectos más cercanos a entregarse tengan vacíos a la hora de definir los requerimientos, ya que en muchos casos las reuniones son pocas, o no se toman las opiniones de todos los usuarios finales, además se omiten pruebas profundas por falta de tiempo y documentación del sistema en desarrollo.

Para ello se propone una metodología de gestión de proyectos para desarrollo ágil Programación Extrema, que garantice un proceso sólido y bien definido de planificación y estandarización en los pasos establecidos en la metodología para el desarrollo de los sistemas. Esto permitirá que se tengan documentados los sistemas, mayor interacción de los actores con el desarrollador, claridad en las ideas de requerimientos y sobre todo se cumpla el tiempo establecido en la planificación.

Antecedentes

En 1994, en la Dirección General de Ingresos no existía una División de Informática y Sistemas, sino que era un área de informática que dependía directamente de la División Administrativa Financiera. Es hasta 1995 que se constituye como una División de Informática con un presupuesto propio y con autoridad de tomas de decisiones a través de los niveles jerárquicos competentes.

A medida que las necesidades surgen y la tecnología avanza, la Dirección Superior en conjunto con la División de Informática deciden apoyarse de las tecnologías de información y empiezan un plan estratégico para la creación de sistemas informáticos que ayuden a una mejora continua en la gestión de los procesos, orientados a un incremento en la recaudación, así como una lucha contra la brecha de la evasión fiscal.

En el año 2006 surge el sistema web denominado Ventanilla Electrónica Tributaria (VET), un proyecto modelo que permite que los contribuyentes realicen las declaraciones en línea, esto redujo notablemente las colas de espera y aumentó la recaudación de impuestos.

En 2016, en vista que la cantidad de sistemas ya eran numerosos y las necesidades en las mejoras de los procesos fiscales con ayuda de la tecnología son latentes, se desarrolló una tesis por el Ing. Gabriel Alonso González titulada “Propuesta de aplicación de la metodología de desarrollo SCRUM en la División de Informática y Sistemas”, cuya implementación no se llevó a cabo.

En 2017 se capacitó al personal de informática en COBIT 5, con la idea de implementarlo en los procesos informáticos de la División de Informática, pero aún no se implementa este poderoso marco de trabajo.

Planteamiento del problema

A medida que la tecnología avanza las oportunidades de mejora en los procesos son más alcanzables, hoy en día es indispensable apoyar con la tecnología a la actividad primaria de la cadena de valor de la Dirección General de Ingresos, ya que permite mantener una recaudación en cumplimiento e incremento de las metas y reduce la brecha de la evasión fiscal al tener sistemas que bloqueen las actividades de los contribuyentes en caso de no estar solvente.

En la División de Informática actualmente hay una gran cantidad de sistemas en uso productivo tanto interno como de lado del contribuyente, así mismo, hay una gran cantidad de proyectos en desarrollo de forma simultánea, muchos con plazos extremadamente cortos que van de una semana a dos meses máximo, esto hace que algunos proyectos más cercanos a entregarse tengan vacíos a la hora de definir los requerimientos, ya que en muchos casos las reuniones son poco frecuentes para el análisis de requerimiento y aclaración de ideas, o no se toman las opiniones de todos los usuarios finales, además se omiten pruebas profundas.

Uno de los problemas más frecuentes en las áreas informáticas y en particular en la División de Informática y Sistemas de la DGI es que en el desarrollo de sistemas se omite la fase de diseño y documentación en general.

Por otro lado se está perdiendo la oportunidad de desarrollo de nuevos proyectos por los retrasos que implica desarrollar cambios después de haber terminado el desarrollo, ya que en muchos casos no queda de acuerdo a lo que normativa quiere expresar en el requerimiento.

Por último en la oficina el proceso de desarrollo se realiza de varias maneras, es decir que el proceso no está bien definido y varía de acuerdo a los equipos de trabajos, áreas o proyectos, esto hace que la estimación del tiempo de desarrollo sea diferente por naturaleza del proyecto.

Justificación

El manejo de sistemas informáticos es ahora un requisito para mejorar la gestión de la información en la Dirección General de Ingresos, para proporcionar una mejor ejecución de los procesos de control interno y atención a los contribuyentes del día a día. El presente estudio tiene la finalidad de proponer una mejora en el proceso de gestión de los proyectos de desarrollo informático de la Unidad de Sistemas Tributarios de la Dirección General de Ingresos, mediante la implementación de la metodología de desarrollo ágil Programación Extrema, que permitirá que la gestión de los proyectos de desarrollo de sistemas se haga de una manera más fácil, ágil, planificada y estándar en todos los sistemas, además que estén documentados.

La metodología favorecerá al analista programador, ya que establece en la planificación reuniones constantes con los involucrados para una mejor comprensión de los requerimientos, así mismo al tener un estándar certificado se formará con buenas prácticas en su estilo de trabajo. Del mismo modo los usuarios y normativa expresarán mejor las ideas de requerimiento que da lugar a menos vacíos en la comprensión y por lo tanto el desarrollo una vez terminado no estará tan alejado a lo que se pide y la etapa de pruebas o ajustes será menor.

Todos estos beneficios se giran a una mejor gestión de los proyectos de desarrollo y mejoras a los procesos internos que favorecen sin duda al giro principal de la Institución.

Objetivos

Objetivo General

Desarrollar la propuesta de implementación de la metodología de Programación Extrema en la gestión de los sistemas de información en la Unidad de Sistemas Tributarios de la Dirección General Ingresos.

Objetivos específicos

1. Realizar un diagnóstico actual en la gestión de los sistemas de información en la DGI.
2. Establecer el flujo de trabajo para la implementación de la metodología de Programación Extrema en la gestión de sistemas de información.
3. Analizar los costos de implementación de la metodología de Programación Extrema en la gestión de sistemas de información.

Capítulo I.

Marco Teórico.

Marco teórico

1. Sistemas de información.

Los sistemas de información (SI) están cambiando la forma que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos de las empresas, proporcionan información de apoyo al proceso de toma de decisiones y, lo que es más importante, facilitan el logro de ventajas competitivas a través de su implementación en las empresas.

Un sistema de información está constituido por personas, datos que generan información en una organización, e incluso pueden incluir procesos manuales y automatizados. Un sistema de información realiza cuatro actividades básicas que son: entrada, almacenamiento, procesamiento y salida de información.

Según Laudon & Laudon (2012) ve un sistema de información como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización. Además de apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores del conocimiento a analizar problemas, visualizar temas complejos y crear nuevos productos. (pág. 15)

Los sistemas de información contienen datos sobre personas, lugares y cosas importantes dentro de la organización, o en el entorno que la rodea. Por información se hace referencia a los datos que se han modelado en una forma significativa y útil para los seres humanos. Por el contrario, los datos son flujos de elementos en bruto que representan los eventos que ocurren en las organizaciones o en el entorno físico antes de ordenarlos e interpretarlos en una forma que las personas puedan comprender y usar.

Según Laudon & Laudon (2012) *La introducción de un nuevo sistema de información implica mucho más que nuevo hardware y software. También implica cambios en los trabajos, habilidades, administración y organización. Al diseñar un nuevo sistema de información, se rediseña la organización. Los creadores de sistemas deben comprender la forma en que un sistema afectará a los procesos de negocios específicos y a la organización como un todo. (pág. 489)*

1.1. Tipos de sistemas de información.

Debido a que el principal uso que se da a los SI es el de optimizar el desarrollo de las actividades de una organización con el fin de ser más productivos y obtener ventajas competitivas, *puesto que hay distintos intereses, especialidades y niveles en una organización, hay distintos tipos de sistemas.*

Ningún sistema individual puede proveer toda la información que necesita una organización.

1.2. Sistemas de información y procesamiento de datos.

Los sistemas de Información de procesamiento de datos, son un conjunto de métodos y procedimientos que permiten obtener información, estos métodos pueden ser distinguidos entre manuales, electromecánicos y electrónicos. *Según Heredero (2008) son los sistemas básicos de negocio que dan servicio al nivel operativo de la organización. Es un sistema computarizado que efectúa y registra las transacciones diarias para dirigir negocios (pág. 113.)*

1.3. Procesos de sistemas de información.

El propósito global de los sistemas de información es ayudar a los empleados de una empresa a obtener y utilizar la información, a comunicarse con otras personas dentro y fuera de la empresa y tomar decisiones eficaces.

Un sistema de información realiza cuatro actividades básicas:

- Entrada de información: proceso en el cual el sistema toma los datos que requiere.
- Almacenamiento de información: puede hacerse por computadora o archivos físicos para conservar la información.
- Procesamiento de la información: permite la transformación de los datos fuente en información que puede ser utilizada para la toma de decisiones
- Salida de información: es la capacidad del sistema para producir la información procesada o sacar los datos de entrada al exterior.

Según Valencia (2015) Los sistemas de información en las organizaciones capturan y administran datos para producir información útil que respalda a la misma organización y sus empleados, clientes, proveedores y socios. Muchas organizaciones consideran que los sistemas de información son esenciales para su capacidad de competir u obtener una ventaja competitiva.

2. Metodología de Programación Extrema (XP)

Es una Metodología ligera de desarrollo de aplicaciones que se basa en la simplicidad, la comunicación y la realimentación del código desarrollado.

La programación extrema (XP) es el enfoque más utilizado del desarrollo de software ágil. Aunque las primeras actividades con las ideas y los métodos asociados a XP ocurrieron al final de la década de 1980 Una variante de XP llamada XP industrial [IXP] se propuso en una época más reciente. IXP mejora la XP y tiene como objetivo el proceso ágil para ser usado específicamente en organizaciones grandes.

2.1. Valores XP

XP define un conjunto de valores que establecen el fundamento para todo trabajo realizado como parte de XP. Cada uno de estos valores se usa como un motor para actividades, acciones y tareas específicas de XP. Los valores son los siguientes:

- ✓ Comunicación.
- ✓ Simplicidad.
- ✓ Retroalimentación.
- ✓ Valentía.
- ✓ Respeto.

2.2. Características

- ✓ Metodología basada en prueba y error para obtener un software que funcione realmente.
- ✓ Fundamentada en principios.
- ✓ Está orientada hacia quien produce y usa software (el cliente participa muy activamente).
- ✓ Reduce el coste del cambio en todas las etapas del ciclo de vida del sistema.
- ✓ Combina las que han demostrado ser las mejores prácticas para desarrollar software, y las lleva al extremo.
- ✓ Cliente bien definido.
- ✓ Los requisitos pueden cambiar.
- ✓ Grupo pequeño y muy integrado.
- ✓ Equipo con formación elevada y capacidad de aprender.

2.3. Roles De La Metodología XP

En (Erljman Piwen & Goyen Fros, 2001) La Propuesta original de Beck incluye los siguientes roles:

Programador

Es el Responsable de implementar las historias de usuario por el cliente. Además, estima el tiempo de desarrollo de cada historia de usuario para que el cliente pueda asignarle prioridad dentro de la iteración. Cada iteración incorpora nueva funcionalidad de acuerdo a las prioridades establecidas por el cliente. El Programador también es responsable de diseñar y ejecutar los test de unidad del código que ha implementado o modificado.

Cliente

Determina la funcionalidad que se pretende en cada iteración y define las prioridades de implementación según el valor de negocio que aporta cada historia. El Cliente también es responsable de diseñar y ejecutar los test de aceptación.

Encargado de pruebas (tester)

Es el encargado de ejecutar las pruebas regularmente, difunde los resultados dentro del equipo y es también el responsable de las herramientas de soporte para pruebas.

Encargado de seguimiento (tracker)

Una de las tareas más importante del tracker, consiste en seguir la evolución de las estimaciones realizadas por los programadores y compararlas con el tiempo real de desarrollo. De esta forma, puede brindar información estadística en lo que refiere a la calidad de las estimaciones para que puedan ser mejoradas.

Entrenador (coach)

Es Responsable del proceso en general. Se encarga de iniciar y de guiar a las personas del equipo en poner en marcha cada una de las prácticas de la metodología XP.

Consultor

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

Gestor (Big Boss)

Es el vínculo entre el cliente y programadores. Experto en tecnología y labores de gestión. Construye el plantel del equipo, obtiene los recursos necesarios y maneja los problemas que se generan. Administra a su vez las reuniones (planes de iteración, agenda de compromisos, etc). Su labor fundamental es de coordinación.

2.4. Fases de la Metodología XP

La programación extrema usa un enfoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro fases: planeación, diseño, codificación y pruebas.

2.4.1. Planeación

La actividad de planeación comienza escuchando por medio de las entrevistas la ejecución de las actividades para recabar requerimientos que permitan que los miembros técnicos del equipo XP entiendan el contexto del negocio para el software y

adquieran la sensibilidad de la salida y características principales y funcionalidad que se requieren.

Escuchar lleva a la creación de historias por parte del usuario, estas son tomadas por los desarrolladores para modelar los requisitos (Pressman, R. 2010).

Los clientes y desarrolladores trabajan juntos para decidir cómo agrupar las historias en la siguiente entrega (el siguiente incremento de software) que desarrollará el equipo XP. Una vez que se llega a un compromiso sobre la entrega (acuerdo sobre las historias por incluir, la fecha de entrega y otros aspectos del proyecto), el equipo XP ordena las historias que serán desarrolladas en una de tres formas:

A medida que avanza el trabajo, el cliente puede agregar historias, cambiar el valor de una ya existente, descomponerlas o eliminarlas. Entonces, el equipo XP reconsidera todas las entregas faltantes y modifica sus planes en consecuencia.

2.4.2. Diseño

El diseño XP sigue rigurosamente el principio MS (mantenlo sencillo). Un diseño sencillo siempre se prefiere sobre una representación más compleja. Además, el diseño guía la implementación de una historia conforme se escribe: nada más y nada menos. Se desalienta el diseño de funcionalidad adicional porque el desarrollador supone que se requerirá después.

XP estimula el uso de las tarjetas CRC como un mecanismo eficaz para pensar en el software en un contexto orientado a objetos. Las tarjetas CRC (clase-responsabilidad-colaborador) identifican y organizan las clases orientadas a objetos que son relevantes para el incremento actual de software. Las tarjetas CRC son el único producto del trabajo de diseño que se genera como parte del proceso XP (Pressman, R. 2010).

2.4.3. Codificación

Un concepto clave durante la actividad de codificación (y uno de los aspectos del que más se habla en la XP) es la programación por parejas. XP recomienda que dos personas trabajen juntas en una estación de trabajo con el objeto de crear código para una historia. A medida que las parejas de programadores terminan su trabajo, el código que desarrollan se integra con el trabajo de los demás. En ciertos casos, esto lo lleva a cabo a diario un equipo de integración. En otros, las parejas de programadores tienen la responsabilidad de la integración. Esta estrategia de “integración continua” ayuda a evitar los problemas de compatibilidad de interfaces y brinda un ambiente de “prueba de humo” que ayuda a descubrir a tiempo los errores (López , P y Francisco, R. s/f.)

2.4.4. Pruebas

La creación de pruebas unitarias antes de que comience la codificación es un elemento clave del enfoque de XP, ya que esto asegura la calidad del software.

Pruebas de aceptación, son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El Cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta que pase correctamente todas las pruebas de aceptación. (Joskowicz, 2008)

2.4.5. Comparativa con la metodología SCRUM

Mientras que podremos describir a Scrum como una metodología de desarrollo de software, XP es más bien, una práctica de programación. Ambas son técnicas ágiles y se utilizan a menudo juntas. En Scrum se describen procesos para identificar y catalogar el trabajo que hay que hacer, dando prioridad a que el trabajo mediante la comunicación con el representante del cliente o el mismo cliente y además la implementación de ese trabajo mediante comunicados iterativos.

No existen tal vez diferencias tan marcadas, e incluso hasta se puede llegar a trabajar con ambas en conjunto, pero aun así, te mostramos cuatro aspectos importantes en cuanto a diferencias entre una y otra metodología.

Los equipos de Scrum suelen trabajar en iteraciones (llamadas sprints) que son a partir de dos semanas a un mes de duración. Los equipos de XP suelen trabajar en iteraciones que son de una o dos semanas de duración.

Los equipos de Scrum no permiten cambios en sus 'sprints'. Una vez que la reunión de planificación del sprint se ha completado y se ha hecho un compromiso contraído con la entrega de un conjunto de elementos del producto, ese conjunto de elementos se mantiene sin cambios hasta el final del sprint. En XP son mucho más sensibles al cambio dentro de sus iteraciones, siempre y cuando el equipo no haya empezado a trabajar en alguna característica particular, una nueva característica de tamaño equivalente se puede intercambiar en iteración del equipo XP a cambio de la función sin empezar.

Equipos de Extreme Programming suelen trabajar en un orden de prioridad estricta. Las características que se desarrollarán son priorizados por el cliente (de Scrum Product Owner) y el equipo que requiere para trabajar en el proyecto. Por el contrario, el equipo Scrum prioriza la acumulación de productos, pero el equipo determina la secuencia en la que se desarrollarán los elementos del backlog. Un equipo Scrum no opta por trabajar en el tema de mayor prioridad y es muy probable que puedan optar por trabajar en la segunda más importante.

Scrum no prescribe ninguna práctica de ingeniería; XP lo hace. En particular en XP, las cosas se van desarrollando en base a pruebas, el enfoque en las pruebas automatizadas, la programación en parejas, diseño simple, refactorización, y así sucesivamente.

3. Ingeniería Web

La ingeniería web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web.

Rodriguez (2009) comenta que la Ingeniería Web (IWeb) aplica “sólidos principios científicos, de ingeniería y de administración, y enfoques disciplinados y sistemáticos para el desarrollo, despliegue y mantenimiento exitoso de sistemas y aplicaciones basados en Web de alta calidad”. Actualmente existen millones de sitios web que cubren diferentes necesidades de la vida, seguramente la comunicación es la más importante teniendo en cuenta que las distancias pasaron a un segundo plano.

Otra definición de la IWEB es la que plantea S. Murugesan, Y. Deshpande, S es el proceso utilizado para crear, implantar y mantener aplicaciones y sistemas Web de alta calidad.

De hecho la WWW (World Wide Web) se ha establecido como una plataforma común para el desarrollo de sistemas de software. Las aplicaciones Web se han generalizado y se encuentran desarrollos importantes en diferentes ámbitos de la vida humana tales como la economía, industria, educación, administración pública, etc.

3.1. Proceso de la Ingeniería Web.

La inmediatez, evolución y crecimiento continuos, son características de las aplicaciones Web, esto nos lleva a un proceso incremental y evolutivo, que permite que el usuario se involucre activamente, facilitando el desarrollo de productos que se ajustan a sus requerimientos.

Según (Pressman 2015) enumera siete actividades que forman parte del proceso de la IWeb y que son aplicables a cualquier aplicación web independientemente de su tamaño y complejidad. (pág. 183) Las mismas son:

1. La **formulación** identifica objetivos y establece el alcance de la primera entrega.
2. La **planificación** genera la estimación del coste general del proyecto, la evaluación de riesgos y el calendario del desarrollo y fechas de entrega.
3. El análisis especifica los requerimientos e identifica el contenido.
4. La **modelización** se compone de dos secuencias paralelas de tareas. Una consiste en el diseño y producción del contenido que forma parte de la aplicación. La otra, en el diseño de la arquitectura, navegación e interfaz de usuario. Es importante destacar la importancia del diseño de la interfaz. Independientemente del valor del contenido y servicios prestados, una buena interfaz mejora la percepción que el usuario tiene de éstos.
5. En la **generación de páginas** se integra contenido, arquitectura, navegación e interfaz para crear estática o dinámicamente el aspecto más visible de la aplicación: las páginas.
6. El **test** busca errores en todos los niveles: contenido, funcional, navegacional, rendimiento, etc. El hecho de que las aplicaciones residan en la red, y que inter operen en plataformas muy distintas, hace que el proceso de test sea especialmente difícil.
7. Finalmente, el resultado es sometido a la evaluación del cliente.

3.2. Diferencia con la Ingeniera de Software.

Una de las principales diferencias es que la Ingeniería Web para el desarrollo de una aplicación web es un esfuerzo multidisciplinario debido al manejo de múltiples formatos, con una mezcla de atención social para saber su efecto ético y legal.

Según Rodríguez (2009) define como Ingeniería de software como la disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad. (pág. 9.)

La Ingeniería Web (IWeb) está relacionada con el establecimiento y utilización de principios científicos, de ingeniería y de gestión, y con enfoques sistemáticos y disciplinados del éxito del desarrollo, empleo y mantenimiento de sistemas y aplicaciones basados en web de alta calidad.

La Ingeniería Web no es un clon de la Ingeniería de Software. El desarrollo de aplicaciones web es distinto respecto al desarrollo de software de otras categorías.

Según Rodríguez (2009) en muchas ocasiones, en el desarrollo de aplicaciones web, se aplican, herramientas de Ingeniería de Software, sin tener en cuenta que contienen características especiales como: usabilidad, navegabilidad, seguridad, mantenimiento, escalabilidad, entre otras. (pág. 13.)

3.3. Tecnología en el desarrollo web.

Gallego Micael (2014) Desarrollo de aplicaciones web, Universidad Rey Juan Carlos, España expresa que: El desarrollo de aplicaciones web ha evolucionado enormemente en la última década, tanto desde el punto de vista del desarrollo de software como a nivel de administración de sistemas. Se han creado multitud de tecnologías, frameworks de desarrollo de aplicaciones, bibliotecas, aplicaciones configurables, arquitecturas, modelos de publicación de versiones (Pág.3.)

Layouts aplicación para CSS

Layouts es un sencillo simple creador de CSS que puede ser usado para crear web, soporta estándares de diseños de página. Layouts se inspiró en Yahoo! 's Grids CSS y posee 5 opciones predefinidas de anchos que soporta ancho tipo líquido o fijo.

Layouts también tiene una web basada en patrones prediseñados que fue desarrollado para el navegador Firefox. Precaución: el diseño de patrones está todavía en sus primeras etapas de desarrollo, por lo que puede generar algunos errores.

Lenguaje del lado del servidor y del cliente.

Adebal de la Torre (2006) afirma que los lenguajes de lado servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él (Pág.1.)

Por otro lado, los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también Java y JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre- tratamiento.

Cada uno de estos tipos tiene por supuesto sus ventajas y sus inconvenientes. Así, por ejemplo, un lenguaje de lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio sin necesidad de pagar más ya que, por regla general, los servidores que aceptan páginas con scripts de lado servidor son en su mayoría de pago o sus prestaciones son muy limitadas.

Lenguajes del lado del servidor

C# es un lenguaje elegante, con seguridad de tipos y orientado a objetos, que permite a los desarrolladores crear una gran variedad de aplicaciones seguras y sólidas que se ejecutan en NET Framework. Puede usar C# para crear aplicaciones cliente de Windows, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos y muchas, muchas más cosas. Visual C# proporciona un editor de código avanzado, prácticos diseñadores de interfaz de usuario, un depurador integrado y muchas otras herramientas que facilitan el desarrollo de aplicaciones basadas en el lenguaje C# y .NET Framework. (Microsoft, 2015)

Active Server Pages (ASP):

Es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). EL estándar ASP permite utilizar cualquier lenguaje para la programación, si bien Microsoft favorece que se utilice el VBScript, y en segundo término el JScript (versión del estándar Javascript hecha por Microsoft). (gallego Micael 2014.)

Lenguajes del lado del cliente.

Según el Instituto Tecnológico de Matehuala, Programación web, México (2012): Los lenguajes de programación del lado cliente se usan para su integración en páginas web. Un código escrito en un lenguaje de script se incorpora directamente dentro de un código HTML y se ejecuta interpretado, no compilado (Pág. 102).

Con la programación del lado del cliente se pueden validar algunos de los datos en la máquina cliente antes de enviarlos al servidor. Esto proporciona a los usuarios informes de error inmediatos, mientras siguen en esa página de formulario y sin necesidad de volver atrás tras recibir un mensaje de error. Puede resultar necesario acceder a una base de datos para validar determinados valores, mientras que no suele disponer de un acceso directo a la base de datos en la máquina del cliente, aunque ese acceso a la base de datos es factible.

Instituto Tecnológico de Matehuala, Programación web, México (2012) expresa que significa que un código escrito en un lenguaje de script no sufre ninguna transformación previa a su ejecución. Cada línea de código es traducida a lenguaje máquina justo antes de su ejecución. Después es ejecutada y la traducción no se conserva en ningún sistema de almacenamiento (como discos, cintas, etc.)

JavaScript:

JavaScript es un lenguaje de scripts compacto basado en objetos (y no orientado a objetos). Originariamente era denominado LiveScript, y fue desarrollado por Netscape para su navegador Netscape Navigator 2.0. Fue éste el primer cliente en incorporarlo. Se ejecuta sobre 16 plataformas diferentes, incluyendo los entornos de Microsoft. (*Instituto Tecnológico de Matehuala, Programación web, México ,2012.*)

Por ejemplo, una aplicación escrita en JavaScript puede ser incrustada en un documento HTML proporcionando un mecanismo para la detección y tratamiento de eventos, como pulsación del ratón o validación de entradas realizadas en formularios. Sin existir comunicación a través de la red, una página HTML con JavaScript incrustado puede interpretar, y alertar al usuario con una ventana de diálogo, de que las entradas de los formularios no son válidas. O bien realizar algún tipo de acción como ejecutar un fichero de sonido, un applet de Java, etc. (*Instituto Tecnológico de Matehuala, Programación web, México ,2012.*)

Motores de bases de datos.

Yomar Altamar (2013), expresa que: Los Motores de Bases de Datos, son programas específicos, dedicados a servir de intermediarios entre las bases de datos y las aplicaciones que las utilizan como función principal, cada uno de estos cumple una tarea específica, que van desde crear la Base de Datos hasta administrar el uso y acceso a ésta (Pg. 9.)

Un MBD está compuesto principalmente por tres lenguajes, siendo estos DDL (Lenguaje de definición de datos), DML (Lenguaje de manipulación de datos), y finalmente el SQL (Lenguaje de consulta.)

Principal motor de bases de datos.

Microsoft SQL Server es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos.

Entornos de desarrollo integrados.

Microsoft Visual Studio 2012 es un conjunto de herramientas para crear software, desde la fase de diseño pasando por las fases de diseño de la interfaz de usuario, codificación, pruebas, depuración, análisis de la calidad y el rendimiento del código, implementación en los clientes y recopilación de telemetría de uso. Estas herramientas están diseñadas para trabajar juntas de la forma más eficiente posible y todas se exponen a través del Entorno de desarrollo integrado (IDE) de Visual Studio. (Microsoft, 2013)

3.4. Ingeniería web basada en UML (UWE.)

3.4.1. Modelos.

Según Cachero Castro (2008), UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito. OO-H: Una extensión de los métodos OO para el modelado y generación automática de interfaces hipermedia (pg. 28.)

Según Schwabe G. Rossi (2008), es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado. UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la

definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario. Desarrollo de Aplicaciones Hipermedia con OOADM. Taller sobre el Proceso de Desarrollo de Hipermedia, Métodos y Modelos, Hypertext98, Pittsburg, EE.UU (Pg. 31.)

Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de una meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos.

El método y sus seis modelos:

1. Modelo de casos de uso para capturar los requisitos del sistema.
2. Modelo conceptual para el contenido (modelo del dominio.)
3. Modelo de usuario: Modelo de navegación que incluye modelos estáticos y dinámicos.
4. Modelo de estructura de presentación, modelo de flujo de presentación.
5. Modelo abstracto de interfaz de usuario y modelo de ciclo de vida del objeto.
6. Modelo de adaptación.

3.4.1.1. Modelo de casos de usos.

De acuerdo con la Universidad Nacional de la Plata (2009), La calidad de datos es considerada un aspecto importante en relación con el éxito o fracaso de las tareas cotidianas en una organización. Hoy en día, la mayoría de estas tareas se encuentran soportadas por aplicaciones de software, *Metodologías de diseño usadas en ingeniería web, su vinculación con las NTICS (Pg. 75).*

La especificación temprana de los requisitos que deben cumplir esas aplicaciones es un desafío para la Ingeniería del Software.

Dichos requisitos serán capturados desde modelos de procesos de negocio descritos con BPMN (Business Process Model and Notation), y expresados mediante casos de uso de UML (Unified Model Language.)

Con este propósito se propone un método mediante el cual, de manera ordenada y sistemática, los expertos del negocio pueden modelar procesos de negocio consciente de la calidad de datos y obtener desde dichos modelos artefactos útiles para la creación de software.

Según Hammon & Wolf(2011) y Recker, (2010), los lenguajes más usados para el modelado de BP son UML y BPMN, siendo este último el más utilizado en la industria por lo que es ampliamente reconocido como el estándar de facto para el modelado de BP.

Bocciarelli & D'Ambrogio(2011), entre las extensiones propuestas en la literatura se pueden mencionar algunas orientadas a: expresar algunas necesidades de los clientes, tales como tiempo, costo y fiabilidad, especificar propiedades no funcionales como desempeño y fiabilidad, así mismo modelar requisitos de seguridad en BP, representar explícitamente restricciones legales (Goldner&Papproth, 2011), analizar el desempeño de los procesos de negocio (Lodhi, Veit, &Saake, 2011) y expresar requisitos de calidad de datos, entre otras.(Págs. 19-22.)

3.4.1.2. Modelo conceptual para el contenido.

Un concepto para este caso, en términos de la Programación Orientada a Objetos, es un objeto del mundo real; es decir, es la representación de cosas del mundo real y no de componentes de software.

Informalmente se puede decir que un concepto es una idea, cosa u objeto. Para descubrirlos se debe analizar los sustantivos en las descripciones textuales del dominio del problema, es decir, de la descripción del sistema, de los requerimientos y de los casos de uso.

3.4.1.3. Modelo de usuario

Según Bocciarelli & D'Ambrogio (2011), la arquitectura de la interfaz de usuario se define teniendo en cuenta diferentes aspectos que introducen modelos para soportar el modelado de rol usuario, el modelado de contexto, el modelado de tareas, el modelado de dominio y los modelos de presentación. Nuestro principal interés en el desarrollo de la interfaz de usuario se centra en los modelos de presentación de la interfaz de usuario como factor esencial incluso para desarrollo basado en modelos de la interfaz de usuario de sistemas propiedades no funcionales como desempeño y fiabilidad (Pg.24.)

Se definen cuatro etapas para la presentación de la interfaz de usuario: en la primera etapa, el modelo de contexto de la interfaz de usuario que define los componentes de la presentación de la interfaz de usuario en una forma menos detallada y de manera general. Cada etapa añade información más detallada hasta llegar a una interfaz de usuario final.

Un modelo de dominio especifica los datos que los usuarios manipulan acerca de las entidades del mundo real y sus interacciones tal como son entendidas por los usuarios en relación con las acciones que son posibles sobre estos conceptos.

De acuerdo con J. William & C. Collazos (2009), la arquitectura de la interfaz de usuario se define teniendo en cuenta diferentes aspectos que introducen modelos para soportar el modelado de rol-usuario, el modelado de contexto, el modelado de tareas, el modelado de dominio y los modelos de presentación. (Pg. 198.)

Se usarán los diagramas de clases UML como la base para expresar un modelo de dominio, ya que es el estándar de facto en modelado de diagramas de clases.

Un modelo de usuario se compone de una lista de roles de usuario identificados. Un rol de usuario es un conjunto de usuarios que comparten características similares y que pueden ser dispuestos de forma jerárquica. La información del usuario representa un conjunto de intereses comunes, sus conductas y responsabilidades.

Un modelo de usuario se define por la comprensión del uso del sistema y son muy indicados para alcanzar fines de usabilidad. Ellos identifican y representan los aspectos esenciales de las necesidades y requerimientos de los usuarios.

3.4.1.4. Modelo de estructura de presentación.

Según Cristina Cachero Castro (2008), OO-H: *Una extensión de los métodos OO para el modelado y generación automática de interfaces hipermedia.* 'Un diagrama de estructura compuesta es un diagrama que muestra la estructura interna de un clasificador, incluyendo sus puntos de interacción a otras partes del sistema". Esto muestra la configuración y relación de las partes que juntas realizan el comportamiento de clasificador contenido (Pg. 14.)

3.4.1.5. Modelo abstracto de interfaz

Diseño de interfaz abstracta

J. William & C. Collazos (2009), Las interfaces de usuario abstractas modelan información que es independiente de la modalidad en la cual será implementada la interfaz. Estas corresponden a modelos que son independientes de la tecnología en la cual se decide realizar la interfaz de usuario final.

(Pg.204.)

Una separación ordenada entre ambas preocupaciones, de navegación y diseño de interfaz abstracta, permite construir interfaces diferentes para el mismo modelo de navegación, llevando a un grado más alto de independencia de tecnología de la interfaz de usuario. En suma, esta separación permite entender mejor la aplicación global de la estructura para indicar qué transformaciones claramente en la interfaz serán transformaciones navegacionales.

En OOHDM, se usa un acercamiento del diseño de datos de vista abstractos (ADVs), para describir la interfaz del usuario de una aplicación de hipertexto. Los ADVs son objetos en los que tienen un estado y una interfaz, donde la interfaz puede ser ejercido a través de mensajes (en particular, eventos externos generados por el usuario).

Las ADVs son abstractas en el sentido de que ellos sólo representan la interfaz y el estado, y no la aplicación. Las ADVs han sido usados para representar interfaces entre dos medios de comunicación diferentes como un usuario, una red o un dispositivo (un cronómetro, por ejemplo) o como una interfaz entre dos o más objetos de datos abstractos (ADOs.)

Los ADOs son objetos que no soportan externamente eventos generados por el usuario [Cowan 95]. Desde un punto de vista arquitectónico, los ADVs son observadores para ADOs, para que el protocolo de comunicación entre la interfaz y los objetos de aplicación siga las reglas descritas en el modelo de diseño de observador [Gamma 95].

Un ADV usado en el diseño de aplicaciones web puede verse como un objeto de interfaz. Comprende un conjunto de atributos (y objetos de interfaz anidado) que define sus propiedades de percepción, y el conjunto de eventos que puede manejar, como eventos generados por el usuario. Los ejemplos de eventos generados por el usuario son MouseClick, MouseDoubleClick, MouseOn, etc.

De acuerdo con Schwabe, G. Rossi (2008). Pueden definirse valores del atributo como constantes y pueden definirse estilos particulares de apariencia como posición, color, o sonido. Los modelos de interfaz ADV unen al modelo que permite tratar estos rasgos de una manera abstracta y los relega al paso de la aplicación. Las ADVs pueden ser fácilmente implementadas en ambientes orientados a objetos para el web o puede traducirse a documentos HTML (Pg.17.)

3.4.2. Ventajas del uso del UWE.

Una de las ventajas que UWE extiende el estándar UML es la flexibilidad de éste para la definición de un lenguaje de modelado específico para el dominio web y sobretodo la aceptación universal de dicho estándar en el campo de la Ingeniería del Software.

Permite especificar de mejor manera una aplicación web en su proceso de creación mantiene una notación estándar basada en el uso de UML (Unified Modeling Language para sus modelos y sus métodos, lo que facilita la transición.

La metodología define claramente la construcción de cada uno de los elementos del modelo y mantiene una notación estándar basada en el uso de UML (Unified Modeling), además de almacenar las propiedades que describen a los diferentes tipos de usuarios.

Otra gran ventaja es que actualmente existen múltiples herramientas CASE basadas en UML, con lo cual es relativamente sencillo su utilización y ampliación para utilizar los objetos de modelado definidos en UWE. Éste, hace uso de notación UML pura y los tipos de diagramas UML en donde sea posible para el análisis y diseño de aplicaciones web. En adición, la extensión de UWE cubre la navegación, presentación, lógica del negocio y aspectos de adaptación. La notación UWE se define como una extensión "ligera" de UML.

Capítulo II.

Desarrollo y justificación de la solución propuesta.

Aspectos organizacionales

La Dirección General de Ingresos DGI, nace el día 29 de junio de 1957, mediante la publicación del Decreto No. 243 "Ley creadora de la Dirección General de Ingresos", en La Gaceta, Diario Oficial No. 144 y la Ley Creadora de la Dirección General de Aduanas y de reforma a la Ley creadora de la Dirección General de Ingresos publicada en la gaceta No. 69 del 6 de abril del dos mil dos.

Es una institución descentralizada con autonomía administrativa y financiera, cuyo objeto es aplicar y hacer cumplir las Leyes, actos y disposiciones que establecen o regulan ingresos a favor del Estado, para la ejecución de proyectos de interés común en la sociedad.

Visión

Ser una Administración Tributaria moderna, respetada y calificada, al servicio del pueblo nicaragüense.

Misión

Recaudar los tributos internos de manera eficaz, eficiente y transparente, promoviendo la cultura de cumplimiento voluntario, aportando al Gobierno de Reconciliación y Unidad Nacional, recursos para el Plan de Desarrollo Humano y el Programa Económico.

Estructura Organizacional

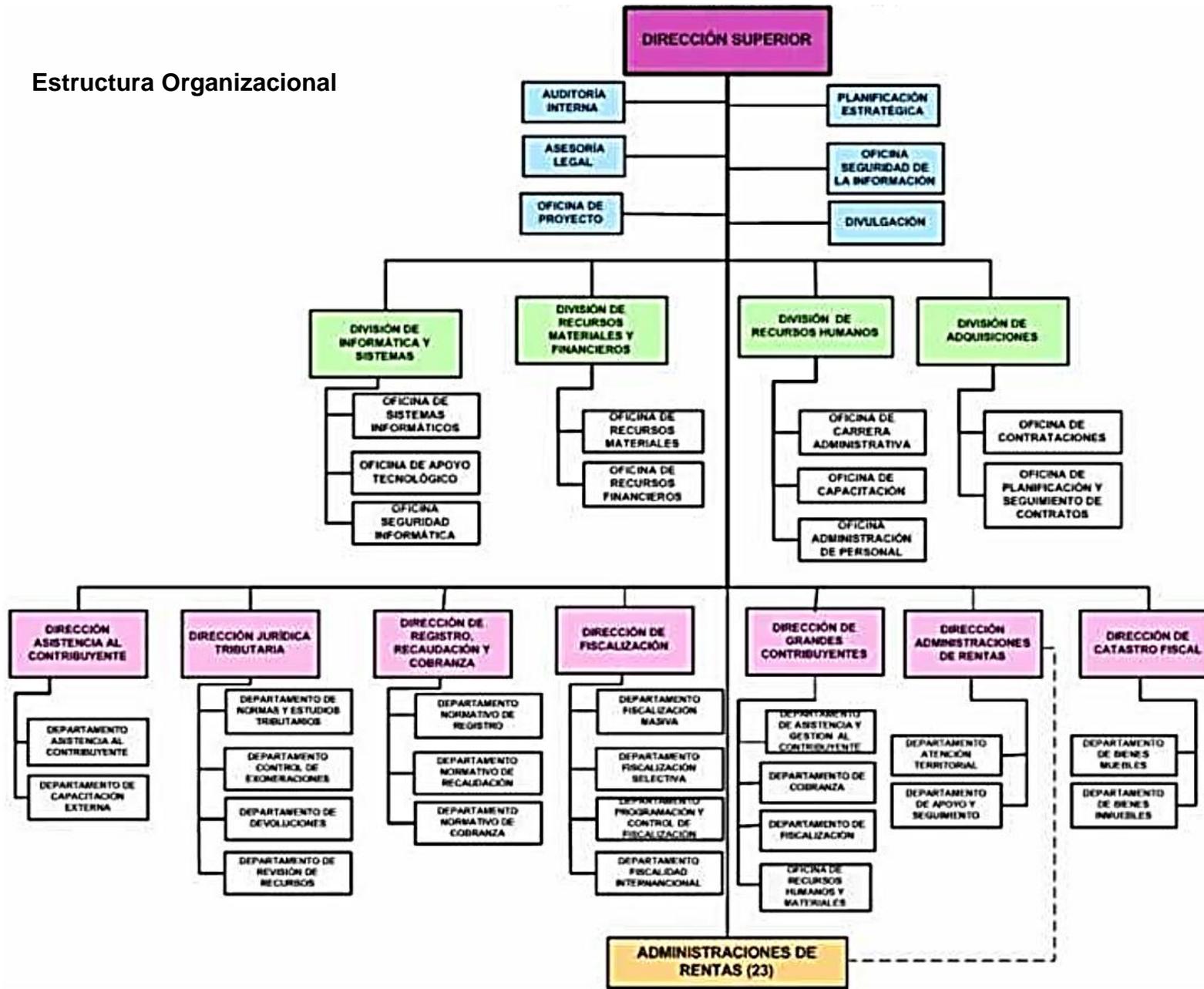


Figura 1. Estructura Organizacional de la Dirección General de Ingresos 2019.

Enfoque sistémico.

Diagrama causa y efecto de la problemática.

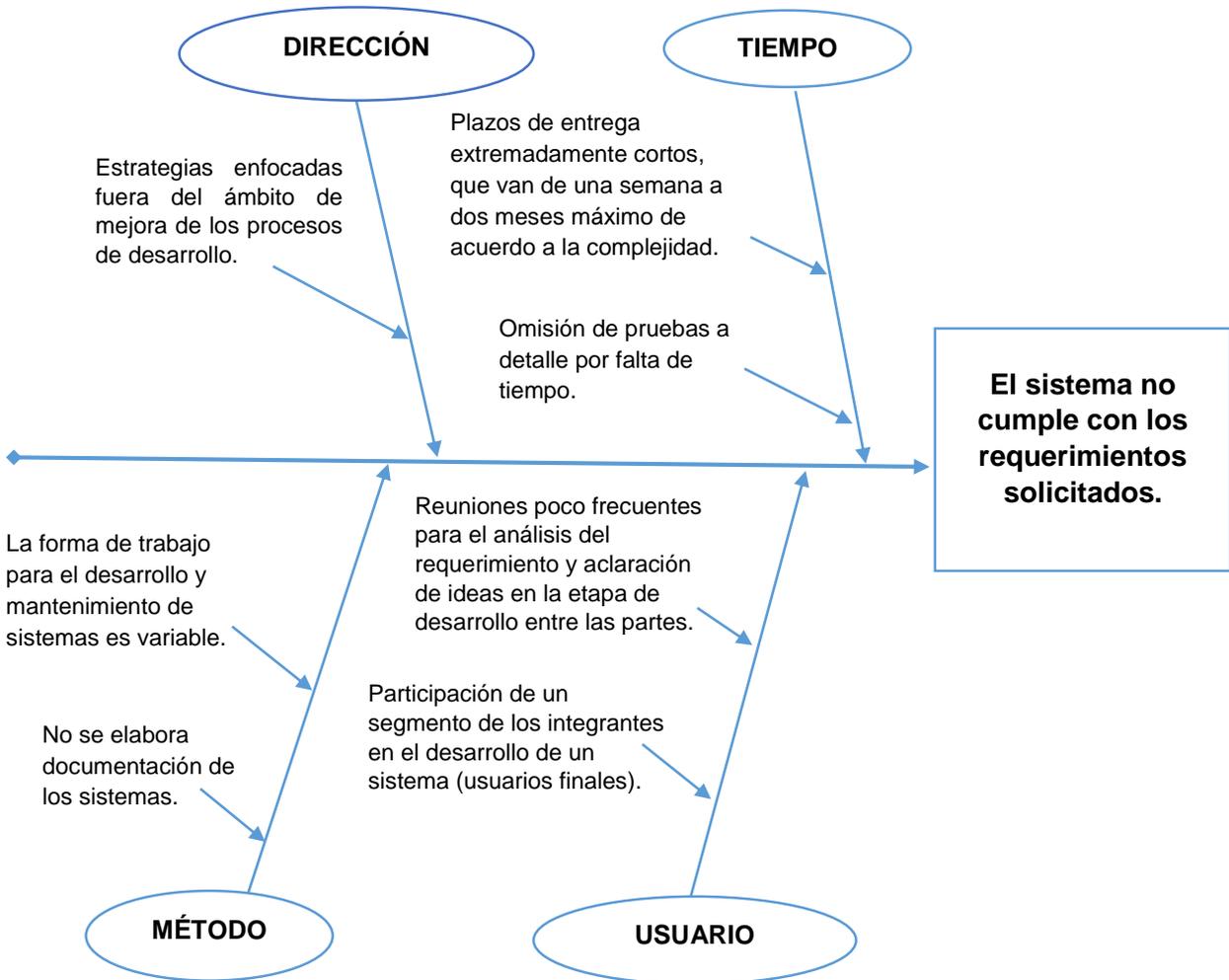


Figura 2. Diagrama de causa y efecto.

Universo

Dirección General de Ingresos

Sistema

Unidad de Sistemas Tributarios de la Dirección General de Ingresos.

Campo de acción

Gestión de proyectos de desarrollo de los sistemas de información en la Unidad de Sistemas Tributarios de la Dirección General de Ingresos.

Objetivos del sistema

- ✓ Analizar, diseñar, programar los procesos automatizados que en las unidades operativas de la Institución según el plan anual.
- ✓ Garantizar el buen funcionamiento de los procesos automatizados a nivel nacional de la institución.
- ✓ Brindar el mantenimiento a los sistemas productivos existentes que muestren inconsistencias y resolver con alta disponibilidad.
- ✓ Adecuar los sistemas existentes a las nuevas necesidades de las diferentes unidades operativas.

Definición de actores.

Los actores que intervienen en la gestión de proyectos de desarrollo de sistemas informáticos son los siguientes:

- ✓ Director de informática.
- ✓ Jefe de Oficina de sistemas informáticos.

- ✓ Jefe de unidad de sistemas tributarios.
- ✓ Analista de sistemas.
- ✓ Analista de Normativa.
- ✓ Analista de control de calidad.

Director de informática:



- ✓ Delega responsabilidades según la naturaleza de la asignación y área competente.
- ✓ Controla y dirige las actividades delegadas a los Jefes de oficinas.
- ✓ Elabora informe de las actividades que realiza cada oficina.
- ✓ Controla los recursos asignados a la división en general.
- ✓ Rinde cuentas a la Dirección Superior de los avances de los proyectos en desarrollo.
- ✓ Toma decisiones acerca de las eventualidades que surgen.

Jefe de Oficina de sistemas informáticos:



- ✓ Delega actividades según la naturaleza de la asignación y unidad competente.
- ✓ Controla y dirige las actividades delegadas a los Jefes de unidades.
- ✓ Elabora informe de las actividades que realiza cada unidad.
- ✓ Rinde cuentas al Director de los avances de los proyectos en desarrollo.

Jefe de unidad de sistemas tributarios:



- ✓ Delega actividades según la naturaleza de la asignación y unidad competente.
- ✓ Controla y dirige las actividades delegadas a cada analista de sistema a cargo.

- ✓ Elabora informe de las actividades que realiza cada analista de sistemas.
- ✓ Rinde cuentas al Jefe de oficina de los avances de los proyectos en desarrollo.

Analista de sistemas:



- ✓ Desarrolla los requerimientos asignados por el jefe de unidad.
- ✓ Brinda el mantenimiento correspondiente a los sistemas existentes asignados.
- ✓ Resuelve bitácoras de inconsistencias reportadas de los sistemas a cargo.
- ✓ Rinde cuentas al Jefe de unidad de los avances de los proyectos en desarrollo.

Analista de Normativa:



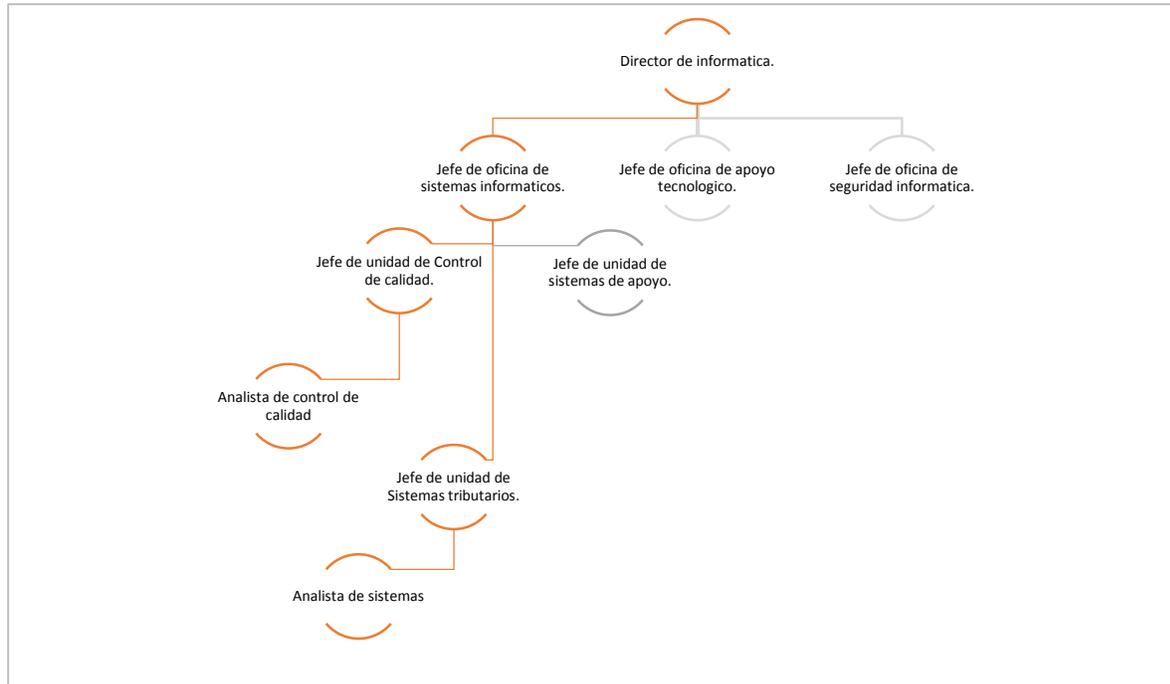
- ✓ Elabora el requerimiento de un nuevo sistema.
- ✓ Realiza pruebas preliminares al sistema.
- ✓ Toma decisiones en conjunto Dirección Superior, de llevar el sistema al ambiente de producción.
- ✓ Brinda aclaraciones al desarrollador del requerimiento a desarrollar.

Analista de control de calidad:



- ✓ Realiza pruebas preliminares al sistema.
- ✓ Aprueba el desarrollo en caso de que el sistema cumpla con lo estipulado en el requerimiento.

Estructura de la línea de mando o jerarquía de puesto.



— Cargos de interés del presente. — Cargos sin interés dentro del presente.

Figura 3. Línea de mando en la División de Informática y sistemas de la DGI.

Proceso actual

Gestión de proyecto de desarrollo de sistemas:

El Director de Informática delega los nuevos sistemas según el plan anual operativo y le proporciona el requerimiento del sistema al Jefe de oficina de sistemas, este recibe los lineamientos e informa y delega al jefe de unidad de sistemas tributarios y le proporciona el requerimiento, a parte establece ciertos lineamientos que considere conveniente, mas lo orientados por la Dirección. El jefe de unidad reúne a su personal encargado para este nuevo desarrollo y prepara las condiciones para que el desarrollador tome el proyecto y entienda las ideas de manera general de el proceso que se quiere automatizar en el sistema y le proporciona una copia del requerimiento.

En este momento se establecen el tiempo de entrega, los permisos de bases de datos, otros aspectos técnicos y los medios de consultas, se realizan reuniones en caso de

que en el requerimiento no establezca bien detallado el proceso o no se entienda la redacción. A continuación se muestra un diagrama del proceso representativo:

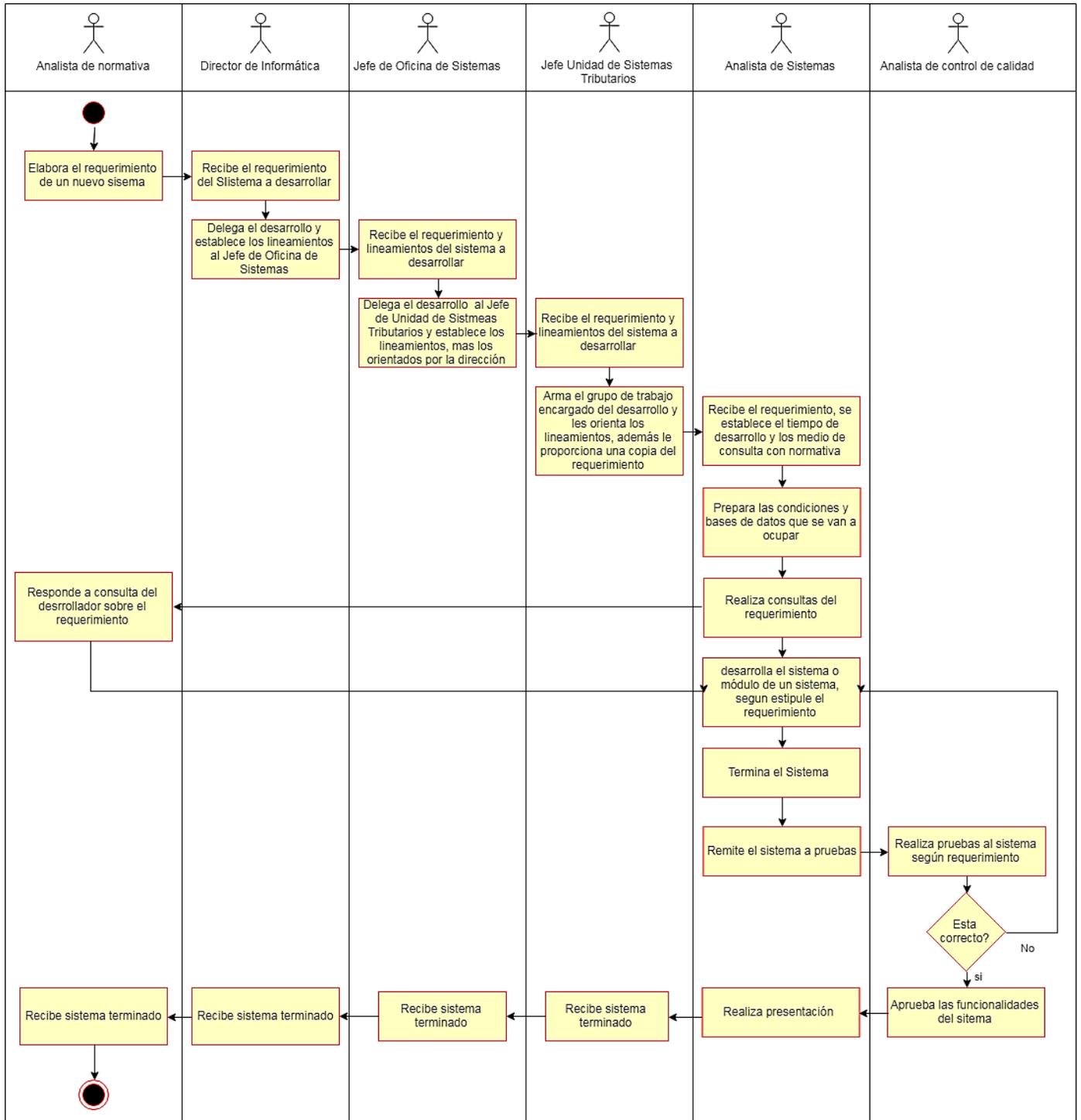


Figura 4. Diagrama de actividad del proceso Gestión de proyecto de desarrollo de sistemas de la unidad de Sistemas Tributarios (UST).

Debilidades en el proceso de gestión de proyecto de desarrollo de sistemas

- ✓ Los requerimientos de los proyectos en desarrollo estan sujeto a cambios.
- ✓ Los tiempos de entrega no estan bien definidos, es decir son demasiado cortos en relacion a lo necesitado en el desarrollo del sistema.
- ✓ El proceso de gestion de proyectos de desarrollo no esta bien definido y hay diferencias entre proyectos.
- ✓ Los usuarios en muchos casos se resisten al cambio.

Resultados de la Entrevista.

De las entrevistas realizadas a las autoridades (**ver anexo 1**) de la División de Informática y Sistemas, se obtuvo que en primer lugar no se implementa actualmente una metodología de desarrollo de sistemas, por lo que la forma de planificar y gestionar los proyectos de desarrollo son variables según el proyecto o el grupo de trabajo.

Se determinó que el proceso de gestión del desarrollo de sistemas es inconsistente, debido en gran parte a que los tiempos de entrega se establecen en plazos extremadamente cortos, los requerimientos son muy cambiantes, no se establece un plan de reuniones, en muchos casos se hacen pruebas en modo superficial y no se elabora documentación.

Conclusiones del diagnóstico.

Debido a que se encontraron inconsistencias en el proceso de gestión del desarrollo de sistemas, es necesario la aplicación de una metodología de desarrollo, y preferiblemente de Programación Extrema, ya que esta metodología cumple con la forma de programar de la División de Informática de la DGI, la cual se caracteriza por tener flexibilidad en el cambio de los requerimientos, desarrollo ágil de los sistemas, organización detallada de las fases del desarrollo y se elabora documentación del sistemas desarrollado.

Propuesta de aplicación de la metodología Programación Extrema (XP)

Introducción a la Metodología

Para la aplicación de la metodología se inicia con la descripción del cliente, en donde se explica el tipo de cliente al que van dirigidos los proyectos de desarrollo, luego se desarrolla la primera fase de la metodología que es la planeación, en esta se definen las historias de usuario, que son los requerimientos del sistema, el plan de entrega, que consta de un cronograma en donde básicamente se establece el periodo de tiempo que va a desarrollarse cada una de las historias además de ver los esfuerzos y prioridad que cada una amerita, y por último en la planeación se conforma el equipo de trabajo con los roles específicos que la metodología establece.

Posteriormente se desarrolla el ciclo de vida del sistema, inicialmente se diseñan algunos diagramas UML en este caso se proponen 3, considerados los más fundamentales, ya que los esfuerzos se encaminan al desarrollo y comunicación continua con todos los actores mediante reuniones diarias.

Luego se organizan y agrupan las historias por iteraciones, tomando en cuenta orden de prioridad, naturaleza de la historia, relación y otros. Cada iteración lleva un proceso secuencial en la que se definen una serie de plantillas que descomponen la historia de usuario a un alto nivel de detalle, esto ayuda a un mejor análisis y comprensión de los requerimientos.

Se inicia la primera iteración con la elaboración de una plantilla de tareas de ingeniería en donde se descompone a nivel de detalle la historia de usuario en tareas específicas, y se desarrollan de manera independiente. Posteriormente se desarrollan las tarjetas CRC (clases, responsabilidad, colaboración) en se plasma a nivel de diseño y análisis todas las clases que crearan en el sistema, colaboración entre ellas y acciones que hará cada una de las clases. Luego se elaboran las pruebas de aceptación en las que una vez terminada la historia por los desarrolladores se remite a las pruebas, a cada una de las historias se les puede practicar varias pruebas, a partir de ellas se determinan los resultados generales de la iteración y por último se registran las

bitácoras de reuniones. Todo esto se repite en la segunda u otras iteraciones en caso que las haya, hasta terminar el sistema.

Descripción del Cliente

La Dirección General de Ingresos mediante la División de Informática y Sistemas desarrolla sus propios sistemas con el objetivo principal de fortalecer con el apoyo de la tecnología los procesos de cara a la cadena primaria de la Institución, que es la recaudación de impuestos para el desarrollo de proyectos de bien común en la sociedad.

Los usuarios de los sistemas desarrollados en la División de Informática pertenecen a dos ámbitos: Internos y Externos.

De lado interno:

Se desarrollan sistemas para la ejecución de las tareas en cada oficina de la estructura organizacional, ya sea que tengan relación a las áreas operativas o áreas administrativas. Es indispensable automatizar los procesos de estas áreas ya que se realiza en menor tiempo y con mayor eficiencia.

De lado externo:

Se desarrollan sistemas para el cliente principal de la Institución, este es el contribuyente, sea persona natural o jurídica, estos se apoyan de los distintos sistemas que la institución les ofrece para realizar con comodidad y facilidad los trámites que desee, con plataformas de alta disponibilidad. Además se tiene una planificación en la infraestructura tecnológica con la filosofía de mejora continua.

Planeación

Es la primera fase de la metodología de Programación Extrema. El equipo de trabajo encargado de un proyecto de sistema a desarrollar, designado por la División de Informática, deberá realizar las historias de usuario para la definición de los requerimientos del sistema de manera que se definan estando presentes los involucrados, estas se desarrollarán bajo la plantilla de historia definida en la

metodología, para cada requerimiento se desarrollará una plantilla de historia. La plantilla es la siguiente:

Tabla 1.

Plantilla de historia de usuario.

Historia de usuario	
Numero:	Usuario
Nombre historia:	
Prioridad en negocio:	Riesgo en desarrollo:
Puntos estimados:	Iteración asignada:
Programador responsable:	
Descripción:	
Observaciones:	

Asignación de roles del proyecto

La División de Informática de la Institución deberá crear un equipo de trabajo con sus distintos roles estimados en la metodología, encargado del desarrollo del nuevo sistema o requerimiento de uno existente. Además se deberá capacitar sobre la metodología de Programación Extrema para su eficiente aplicación.

El equipo de desarrollo del sistema deberá estar integrado por personal con los siguientes roles:

Programador: En la institución, el puesto esta denominado como analista de sistemas y en el equipo será el responsable de desarrollar en el sistema computarizado cada una de las historias de usuario definidas. Además de establecer el tiempo de desarrollo por cada una de las historias.

Cliente: En la Institución hay una área que define los requerimientos, en conjunto con la dirección superior, en este caso será parte de los clientes a como también los futuros usuarios finales a quien estará dirigido el sistema a desarrollar.

Encargado de las pruebas (Tester): Es el encargado de probar el sistema y garantizar el visto bueno de que el sistema funciona de acuerdo a lo establecido en las historias. La división de informática tiene una unidad de control de calidad, en este caso será parte de equipo para hacer las pruebas requeridas del sistema en desarrollo y reportar si lo que se definió en la historias está desarrollado en el sistema.

Encargado de seguimiento: La institución deberá garantizar en el equipo un integrante que se encargue del seguimiento a la evolución del proyecto, el cumplimiento de las actividades frente al tiempo definido.

Entrenador (Coach): Se deberá definir en la Institución un entrenador a cargo del proceso en general, que dé inicio y guie al equipo a poner en marcha el desarrollo del sistema con la metodología de Programación Extrema.

Consultor: En la institución hay consultores, internos especializados en los distintos lenguajes de uso, y dominio de los procesos de giro de negocio de la institución. Se deberá garantizar un integrante que brinde asesorías al equipo según la experiencia, para que el proyecto se desarrolle bajo buenas prácticas tanto en desarrollo como en aplicación del proceso. Deberá estar disponible ante el surgimiento de problemas y guiar al equipo a la solución, para no comprometer el tiempo planeado.

Gestor (Big Boss): Se deberá incorporar al equipo una persona que se encargue de ser el vínculo entre los integrantes del equipo, especialmente entre los desarrolladores y el cliente. Deberá gestionar los recursos y preparar las condiciones para el buen funcionamiento del plan en el desarrollo del proyecto, además es el encargado de administrar las reuniones.

Plan de entrega del proyecto

En el plan de entrega el equipo de trabajo de la división de informática de la DGI realizará un cronograma en el que se detallarán las historias por iteración, estas estarán agrupadas para formar una entrega en una fecha establecida en dicho plan, además estará ordenado por nivel de prioridad. La plantilla básica en el plan de entrega es el siguiente:

Tabla 2.
Plantilla de plan de entrega del proyecto.

Historia	Prioridad	Esfuerzo	Fecha inicio	Fecha fin
Historia1	Alta	2	01-02-2018	02-02-2018
Historia2	Alta	2	03-02-2018	05-02-2018
Historia3	Alta	3	06-02-2018	10-02-2018

Diseño UML

En la metodología de Programación Extrema no se hace mucho énfasis en modelar el desarrollo, ya que lleva una serie de plantillas para el análisis de los requerimientos y llevarlo de manera detallada, esto contribuye a un dominio total entre grupo desarrollador y el cliente de lo que se va a desarrollar en el sistema.

Además una razón principal del porqué no se hace mucho énfasis en el modelado del sistema es porque la metodología de Programación Extrema es para un desarrollo ágil, por lo tanto se recomienda que el equipo formado por la división de informática de la DGI establezca en el proceso de desarrollo algunos diagramas, entre ellos el equipo se recomiendan:

1. Diagrama de clases.
2. Diagramas de actividad.
3. Diagrama de secuencia.

Ciclo de vida del sistema

Iteración

Para una mejor organización de la gestión del proyecto y desarrollo del sistema, la metodología de Programación Extrema propone una organización y agrupación de las historias por iteración. Una iteración de desarrollo es un período de tiempo en el que se realiza un conjunto de funcionalidades determinadas que en el caso de XP corresponden a un conjunto de historias de usuarios.

En las iteraciones el equipo formado de la División de informática de la DGI podrá agrupar las historias según el nivel de prioridad, complejidad, disponibilidad de integrantes y otros aspectos. La plantilla de organización de Iteraciones es la siguiente:

Tabla 3.
Plantilla de iteraciones.

Iteración	Historia	Fecha inicio	Fecha fin
1	Historia1	01-02-2018	02-02-2018
1	Historia2	03-02-2018	05-02-2018
2	Historia3	06-02-2018	10-02-2018

Tareas de Ingeniería

Una vez elaborado el plan de entrega de proyecto y definido las historias e iteraciones del desarrollo del sistema el grupo especializado de la División de informática de la DGI deberá descomponer cada historia de usuario en tareas específicas y concretas. Esto se hace para una mejor organización de la entrega de la historia ya que en el equipo se pueden desarrollar varias historias en paralelo por iteración, esto según la cantidad de programadores asignados en el grupo. Esto se hace necesario ya que entre mayor es la complejidad mayor debe ser el esfuerzo (cantidad de programadores), porque generalmente este tipo de metodologías se aplican a sistemas con periodos cortos pero bien planificados.

Una vez teniendo la plantilla de historias agrupadas por iteración se realiza la plantilla general de tareas por historia, antes de realizar la plantilla de tarea específica.

Tabla 4.

Plantilla general de tareas por historia de usuario.

Iteración	Historia	Nombre de la tarea
1	Historia1	Tarea1
1	Historia1	Tarea2
1	Historia2	Tarea3
2	Historia3	Tarea4

Esto permitirá al grupo de trabajo tener un panorama general de las actividades que se tienen que cumplir por historia y por iteración en el tiempo establecido en el plan de entregas.

Una vez que la División de informática de la división creó la plantilla general de tareas debe crear la plantilla de tarea específica a partir de la general. A continuación se muestra la plantilla propuesta por la metodología de Programación Extrema:

Tabla 5.

Plantilla de tarea de ingeniería.

Tarea de Ingeniería	
Número de Tarea: 1	Número de historia: 1
Nombre de la Tarea: Tarea 1	
Tipo de tarea: Desarrollo	
Fecha inicio:	Fecha fin:
Programador responsable:	
Descripción:	

Elaboración de Tarjetas CRC:

Luego de elaborar las plantillas de tareas de ingeniería es necesario realizar las tarjetas CRC (Class-Responsibility-Collaboration), cuyo objetivo de la misma es hacer, mediante tarjetas, un inventario de las clases que vamos a necesitar para implementar el sistema y la forma en que van a interactuar, además se detalla las responsabilidades de las clases y la colaboración con las otras clases, esto para una mejor comprensión

de la lógica del sistemas en desarrollo. La plantilla que debe hacer el equipo de la División de Informática de la DGI es la siguiente:

Tabla 6.
Tarjeta CRC (Class-Responsibility-Collaboration).

Nombre de la Clase	
Responsabilidad	Colaboración
Guardar...	
Verificar...	Clase2

Pruebas de aceptación

Una vez terminada las tarjetas CRC, el equipo formado por la División de Informática de la DGI deberá definir en una plantilla las pruebas de aceptación de manera general.

Tabla 7.
Lista de pruebas de aceptación de manera general.

Numero de Prueba	Numero de Historia	Nombre de la prueba
1	1	Prueba 1
2	1	Prueba2
3	2	Prueba3

Luego se deberá describir cada prueba de aceptación de manera detallada y específica. A continuación se muestra la plantilla de pruebas de aceptación:

Tabla 8.
Descripción de pruebas de aceptación.

Caso de prueba	
Código:	Numero de historia:
Nombre de historia de usuario:	
Condiciones de ejecución:	
Entradas/pasos de ejecución:	
Resultado esperado:	
Evaluación de la prueba:	

Bosquejos

Ya que la metodología ágil de Programación Extrema tiene como naturaleza crear un gran vínculo entre el equipo de desarrollo y el cliente, entonces propone desarrollar reuniones diarias entre las partes. Es recomendable por parte del cliente definir un bosquejo de las opciones y pantallas que llevará cada una de las historias de usuario.

Es importante que se explique bien y se detalle todas las opciones que se especificaron en las reuniones y que concuerden las historias de usuario elaboradas.

Capturas del avance de la iteración

Para cada iteración por parte del equipo de desarrollo de la División de Informática de la DGI deberá facilitar una garantía a los clientes y demás integrantes, que el avance se da en tiempo y forma, por lo tanto se debe hacer llegar por medio de un manual de usuario con capturas del avance en desarrollo, además de un ambiente en donde puedan realizar pruebas y verificar que el avance ha transcurrido en el tiempo acordado en la planeación y demás plantillas que propone la metodología de Programación Extrema.

Resultados

Los resultados se desarrollan por cada iteración, en esta parte el equipo de desarrollo formado por la división de informática y sistemas deberá evaluar la iteración, los avances según la planeación y el visto bueno o recomendaciones que señale el cliente, si es que las hay.

Bitácora de reuniones

Posteriormente se elaboran las plantillas de resumen de las reuniones realizadas por la iteración en desarrollo. En ella se detalla un resumen de los objetivos que se quieren lograr con la reunión y el resultado que se obtuvo. La plantilla es la siguiente:

Tabla 9.
Bitácora de reuniones.

Bitácora de reunión	
Día	
Horario	
Lugar de encuentro	
Actividades realizadas	
Objetivo	
Resultado	
Participantes	

Este proceso se realiza por cada iteración hasta cumplir con lo estimado en el plan de entregas y las historias que definidas.

Ejemplo de aplicación de la metodología de Programación Extrema

Para ejemplificar se escogió un proceso real del área en estudio de la Dirección General de Ingresos. El proceso a modelar es: “Alertas tributarias”.

Descripción del proceso

Cuando los contribuyentes declaren mediante la VET (Ventanilla Electrónica Tributaria), La Dirección General de Ingresos deberá ejecutar un proceso para evaluar si los RUC o Cédulas declarados son válidos. A los contribuyentes que salgan con información inválida les llegará un correo alertando que la información remitida en la declaración posee inconsistencia y se le adjuntará un link para que el contribuyente pueda corregir los datos erróneos. De esta manera la DGI se asegura que la información de los RUC y/o Cédulas que los contribuyentes declaran es confiable. Además se fortalece el hábito de comunicación entre la Institución y el Contribuyente, de manera que se pueden incluir más procesos por esta misma vía.

Descripción del cliente

Si bien es cierto que el usuario final será el contribuyente y es a quien está dirigido el proceso, en este caso no será el cliente detrás del análisis del desarrollo del sistema, anteriormente se habló de un área Normativa dentro de la estructura orgánica, actualmente es el encargado de la elaboración de los requerimientos, es decir cómo funcionará el sistema a desarrollar, por lo tanto tendrá el rol de cliente y quien estará en las reuniones diarias que la metodología establece.

Planeación

En primera instancia de la planeación se debe conformar el equipo de trabajo, con los roles que la metodología establece:

Asignación de Roles

Tabla 10.
Roles del equipo de trabajo de la DGI.

Integrantes del equipo de trabajo	
Gestor (Big Boss):	Rolando Ortega
Consultor	María Alegría
Entrenador (Coach)	Gabriel Alonso
Encargado de seguimiento	Roger Deshón
Analista de control de calidad (Tester)	Jorge Quintana
Analista de sistemas (programador)	Estefan Menocal
Cliente	Jennifer Dávila

Posteriormente se debe programar una primera reunión para determinación de las historias y saber cómo funcionará el proceso de Alertas Tributarias.

Historias de Usuario

Tabla 11.
Historias de usuarios para el proceso Alertas Tributarias.

Nº	Historias
1	Envío de correos de alerta
2	Corrección de planilla

Plan de entrega

Una vez definidas las historias se realiza el plan de entregas mediante el desarrollo de un cronograma de trabajo.

Tabla 12.
Plan de entrega del proyecto de Alertas Tributarias.

Nº	Historia	Iteración	Prioridad	Esfuerzo	Fecha inicio	Fecha fin
1	Envío de correos de alerta	1	Alta	1	04-03-2019	08-03-2019
2	Corrección de planilla	1	Alta	1	11-03-2019	15-03-2019

En el plan de entregas se define el periodo de duración y la cantidad de programadores responsables (esfuerzo) por historia, además de la prioridad. En este caso solamente se definieron dos historias y se agruparon en una sola iteración.

Diseño UML

Para el análisis y diseño se estimaron como estándar del proceso acompañado de la metodología, la elaboración de tres diagramas fundamentales basados en UML:

Diagrama de Actividad

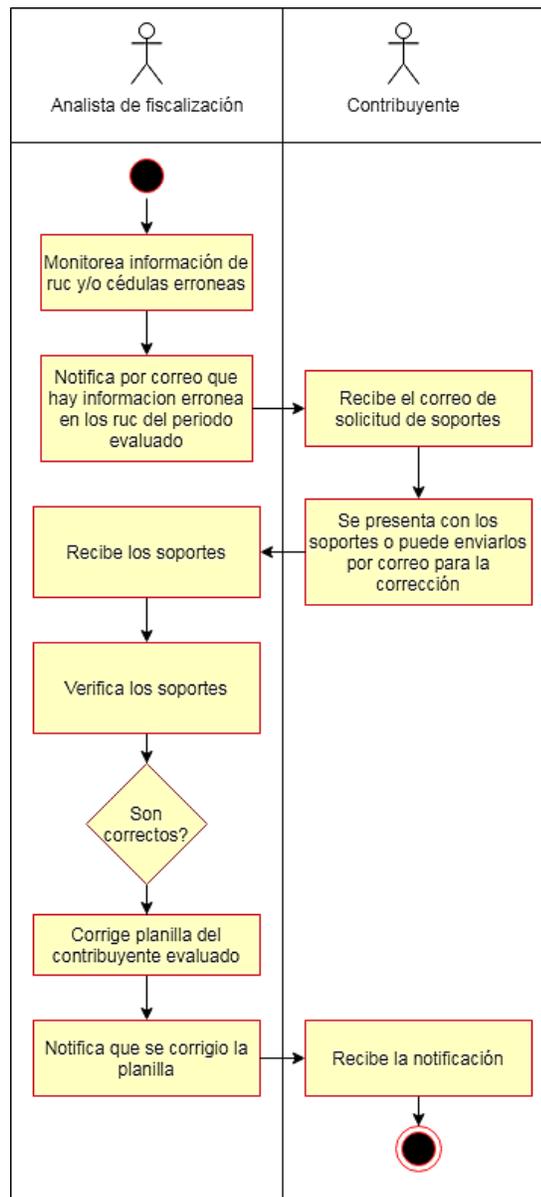


Figura 6. Diagrama de actividad proceso actual de alertas tributarias.

Diagrama de clases

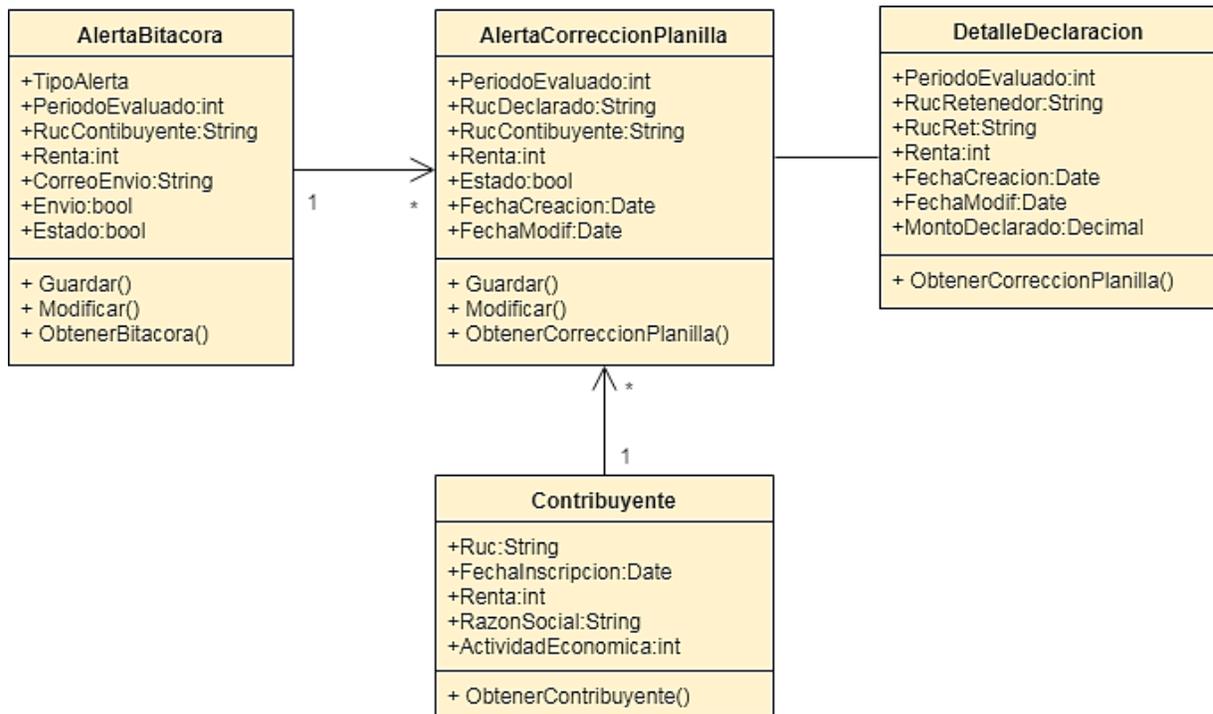


Figura 7. Diagrama de clases del sistema en desarrollo de alertas tributarias.

Diagramas de secuencia

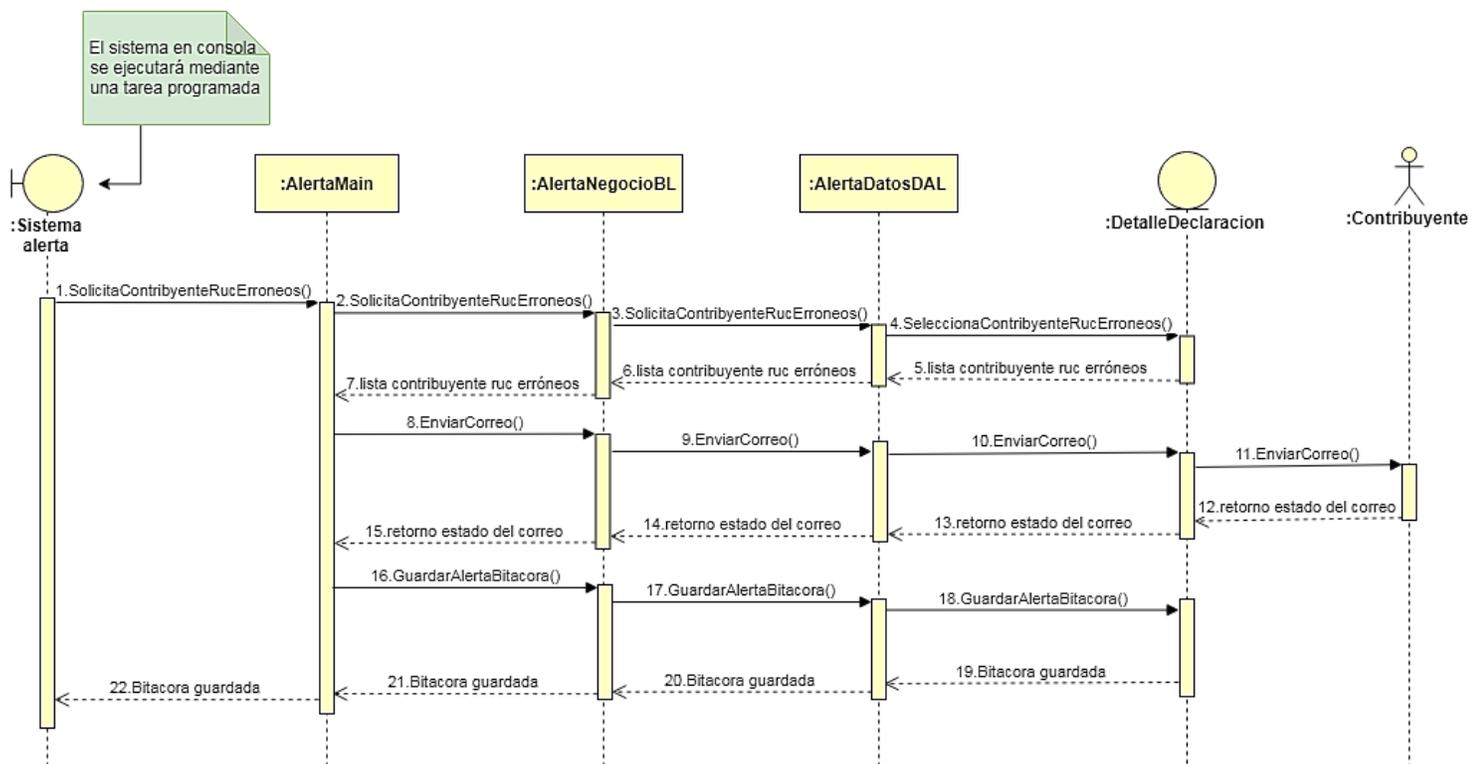


Figura 8. Diagrama de secuencia de envío de correos de alertas tributarias.

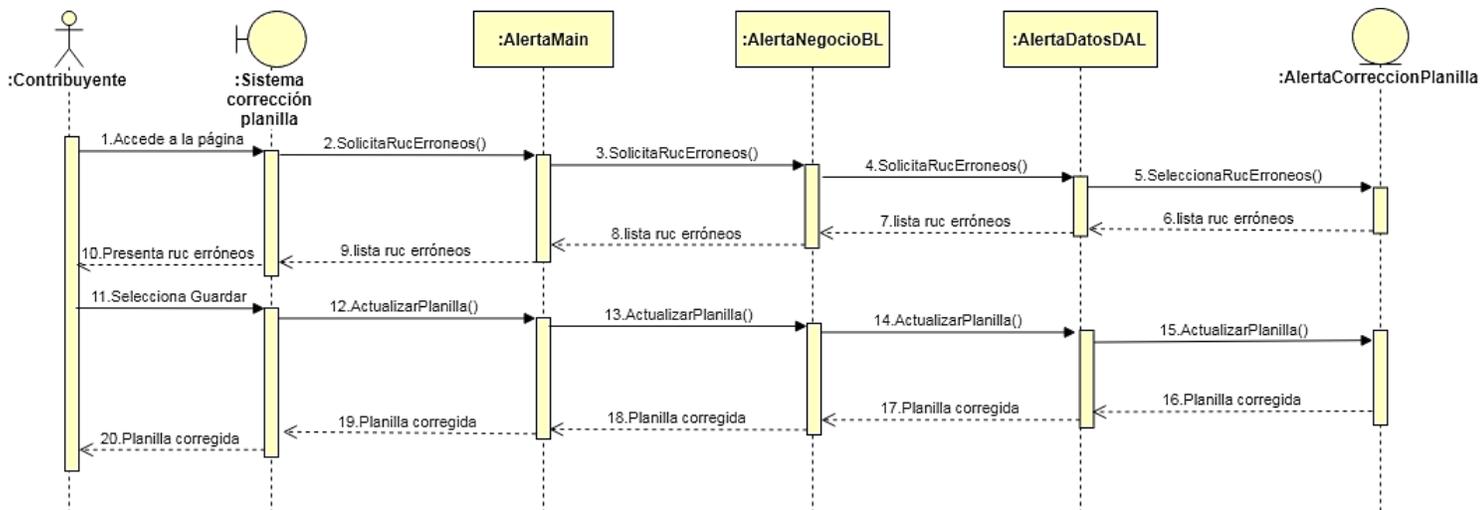


Figura 9. Diagrama de secuencia para corrección de planilla errónea.

Primera Iteración

En la iteración se inicia con la elaboración de la plantilla general de tareas para descomponer cada historia de usuario en actividades específicas.

Tareas de Ingeniería

Tabla 13.
Plantilla general de tareas del proyecto Alertas Tributarias.

Iteración	Historia de usuario		Nombre de la tarea
1	1	Envío de correos de alerta	Programación envío de correo de alerta de índice confiabilidad IVA.
1	1	Envío de correos de alerta	Programación envío de correo de alerta de índice de confiabilidad retenciones.
1	2	corrección de planilla	Programación de corrección de RUC y/o cédulas inexactas
1	2	corrección de planilla	Reenvío de correo de alerta

Luego se elabora las plantillas de Tareas de Ingenierías específicas por cada tarea.

Tabla 14.

Tarea de ingeniería: Programación envío de correo de alerta de índice confiabilidad IVA.

Tarea de Ingeniería	
Número de Tarea: 1	Número de historia: 1
Nombre de la Tarea: Programación envío de correo de alerta índice confiabilidad IVA.	
Tipo de tarea: Desarrollo	
Fecha inicio: 04-03-2019	Fecha fin: 06-03-2019
Programador responsable: Estefan Menocal	
Descripción: Se evaluará la cantidad de contribuyentes que poseen RUC y/o cédulas erróneas en las declaraciones del IVA de un periodo específico y se les enviará un correo con un link con el cual podrán corregir desde un entorno web.	

Tabla 15.

Tarea de ingeniería: Programación envío de correo de alerta de índice de confiabilidad retenciones.

Tarea de Ingeniería	
Número de Tarea: 2	Número de historia: 1
Nombre de la Tarea: Programación envío de correo de alerta de índice de confiabilidad retenciones.	
Tipo de tarea: Desarrollo	
Fecha inicio: 06-03-2019	Fecha fin: 08-03-2019
Programador responsable: Estefan Menocal	
Descripción: Se evaluará la cantidad de contribuyentes que poseen RUC y/o cédulas erróneas en las declaraciones del impuesto de retenciones en la fuente (IR) de un periodo específico y se les enviará un correo con un link con el cual podrán corregir desde un entorno web.	

Tabla 16.

Tarea de ingeniería: Programación de corrección de ruc y/o cédulas inexactas.

Tarea de Ingeniería	
Número de Tarea: 3	Número de historia: 2
Nombre de la Tarea: Programación de corrección de ruc y/o cédulas inexactas	
Tipo de tarea: Desarrollo	
Fecha inicio: 11-03-2019	Fecha fin: 13-03-2019
Programador responsable: Estefan Menocal	
<p>Descripción: Al dar clic al link que se le adjuntará en el correo de alerta lo enviará a una página web en donde se mostrarán los RUC y/o cédulas inexactas que envió el contribuyente en la declaración del periodo evaluado, para corregir y guardar la planilla con la información correcta.</p>	

Tabla 17.

Tarea de ingeniería: Reenvío de correo de alerta.

Tarea de Ingeniería	
Número de Tarea: 4	Número de historia: 2
Nombre de la Tarea: Reenvío de correo de alerta	
Tipo de tarea: Desarrollo	
Fecha inicio: 14-03-2019	Fecha fin: 15-03-2019
Programador responsable: Estefan Menocal	
<p>Descripción: Si el contribuyente corrige su información en la página, pero ingresa un RUC o cédula errónea, la página de corrección envía nuevamente el correo, esta vez con un indicador porcentual de cuantos RUC o cédulas siguen erróneas.</p>	

Tarjetas CRC (Clases, Responsabilidad, Colaboración)

Luego de realizar las tareas de ingeniería se elaboran las tarjetas CRC (Clases, Responsabilidad, Colaboración) en las que se representan las clases que se crearan para el desarrollo del sistema de Alertas Tributarias.

Tabla 18.

Tarjeta CRC AlertaBitacora.

Nombre clase: AlertaBitacora	
Responsabilidad	Colaboración
Guardar Bitácora del envío de correo	AlertaCorreccionPlanilla
Actualizar Bitácora del envío de correo	AlertaCorreccionPlanilla
Verificar ruc/cedulas existentes	

Tabla 19.

Tarjeta CRC AlertaCorreccionPlanilla.

Nombre clase: AlertaCorreccionPlanilla	
Responsabilidad	Colaboración
Guardar AlertaCorreccionPlanilla	Contribuyente, DetalleDeclaracion
Actualizar AlertaCorreccionPlanilla	Contribuyente
Verificar estado de corrección	

Pruebas de aceptación

Luego que se realizan las tarjetas CRC de las clases que se generaron en el desarrollo del sistema, y una vez que se terminó el desarrollo correspondiente a las historias de usuario de la iteración en curso, se remite a pruebas y se elaboran las siguientes plantillas de pruebas de aceptación.

Tabla 20.
Plantilla general de pruebas de aceptación.,

Numero de Prueba	Numero de Historia	Nombre de la prueba
1	1	Prueba de envío de correo
2	2	Prueba de corrección de planilla
3	2	Prueba de reenvío de correo

Posteriormente se elaboras las Descripciones de las pruebas de aceptación de manera específica y detallada.

Tabla 21.
Descripción de la prueba de aceptación: Prueba de envío de correo.

Nombre de la prueba: Prueba de envío de correo	
Código: 1	Número de historia: 1
Nombre de historia de usuario: Envío de correos de alerta	
Condiciones de ejecución: <ul style="list-style-type: none">✓ Disponibilidad del servidor en donde se ejecutará.✓ Acceso al servicio de correo	
Entradas/pasos de ejecución: Ejecutar desde el servidor la aplicación de consola mediante tarea programada de Windows según programación previa para el periodo correspondiente.	
Resultado esperado: <ul style="list-style-type: none">✓ Envío satisfactorio de correo de los contribuyentes que tienen información errónea en las declaraciones del periodo evaluado.✓ Registro de la planilla errónea de los contribuyentes, según la evaluación.	
Evaluación de la prueba: Las pruebas determinaron que el proceso funciona satisfactoriamente	

Tabla 22.

Descripción de la prueba de aceptación: Prueba de Corrección de planilla.

Nombre de la prueba: Prueba de Corrección de planilla	
Código: 2	Número de historia: 2
Nombre de historia de usuario: Corrección de planilla	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El correo haya llegado satisfactoriamente. ✓ El link adjunto funcione correctamente. ✓ El sitio esté disponible. 	
Entradas/pasos de ejecución:	
Hacer clic al link adjunto, corregir los RUC erróneos y hacer clic al botón Guardar.	
Resultado esperado:	
Corrección de los RUC erróneos de la planilla de declaración según periodo evaluado.	
Evaluación de la prueba:	
Las pruebas determinaron que el proceso funciona satisfactoriamente	

Tabla 23.

Descripción de la prueba de aceptación: Prueba de reenvío de correo.

Nombre de la prueba: Prueba de reenvío de correo	
Código: 3	Número de historia: 2
Nombre de historia de usuario: Corrección de planilla	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El correo haya llegado satisfactoriamente. ✓ El link adjunto funcione correctamente. ✓ El sitio esté disponible. 	
Entradas/pasos de ejecución:	
Se envía un nuevo correo en caso que el contribuyente siga ingresando Ruc o Cédula errónea desde la página Web.	
Resultado esperado:	
Envío de correo satisfactoriamente.	
Evaluación de la prueba:	
Las pruebas determinaron que el proceso funciona satisfactoriamente	

Bosquejos del cliente

Se añaden los bosquejos que el cliente pre-diseña y ajunta en el requerimiento antes que se dé la reunión para la presentación y definición de las historias de usuario en la fase de la planeación en la metodología de Programación Extrema.

Avances

Además de la preparación de un ambiente para que el cliente pueda realizar las pruebas preliminares, se adjuntan capturas interactivas del avance de la programación de las dos historias correspondiente a la primera y única iteración.

Resultados de la Iteración

Como resultado de entrega de la primera iteración del sistema de alertas tributarias, normativa realizó sus pruebas preliminares además de las realizadas por control de calidad, y verificó que el desarrollo del sistema corresponde con lo definido en las historias de usuario de la funcionalidad del sistema.

Bitácoras de reuniones

Para terminar la primera iteración se desarrolla una plantilla de bitácora de reuniones, una por cada reunión, a modo de ejemplo se desarrollará dos plantillas, pero en la práctica se deben elaborar todas, según la cantidad de reuniones que sostuvieron.

Tabla 24.

Bitácora de reuniones de la primera iteración.

Bitácora de reunión	
Día	04/03/2019
Horario	3:00 pm - 4:00 pm
Lugar de encuentro	Fiscalización central
Actividades realizadas	Análisis de las historias uno y dos. Propuestas de diseños para exploración de las funciones.
Objetivo	Analizar los puntos requeridos para las historias uno y dos. Determinar los diseños para el funcionamiento de la aplicación.
Resultado	Aclarar para ambas partes la funcionalidad del sistema tanto en proceso, como en diseño y navegación.
Participantes	María Alegría Gabriel Alonso Estefan Menocal Jennifer Dávila

Tabla 25.

Bitácora de reuniones de la primera iteración.

Bitácora de reunión	
Día	06/03/2019
Horario	3:00 pm - 3:30 pm
Lugar de encuentro	Fiscalización central
Actividades realizadas	Realización de Entrevista a analista de normativa.
Objetivo	Recopilar información vital para la elaboración de tareas, tarjetas CRC y programación de pruebas de aceptación de la primera iteración.
Resultado	Recopilar insumos que permitan elaboración de plantillas que la metodología establece, esto para la primera iteración.
Participantes	María Alegría Gabriel Alonso Estefan Menocal Jennifer Dávila

Capítulo III.

Evaluación de costos de la metodología.

Estimación de costos

Para el desarrollo y aplicación de la metodología de Programación Extrema se necesitan algunas condiciones que la metodología propone, por lo tanto para la adquisición de esas condiciones es necesario evaluar los costos totales que se incurrirán.

Equipos disponibles

Dentro de los equipos que la unidad dispone dentro de la DGI, y que se necesita para la aplicación de la metodología de programación extrema están los siguientes:

Tabla 26.
Equipos disponibles.

Equipo	Marca	Modelo
Pc desktop	Dell	Optiplex 790
Impresora multifuncional	HP	DeskJet F4218

La Dirección General de ingresos posee un inventario de estos equipos, de los cuales hay gran cantidad en desuso. Cabe mencionar que para la metodología no se hará contratación, ya que los cargos del personal existente en el área, corresponden a funciones similares al equipo que la metodología establece.

Servicios disponibles

La División de informática cuenta con sus propios servidores dedicados en los cuales aloja todos los sistemas desarrollados incluyendo las bases de datos.

Posee una infraestructura de red intranet (red local) que permite la comunicación entre las áreas, para el cumplimiento de las actividades compartidas, además utiliza un servicio de correo institucional.

Gastos en papelería

Además de los gastos de papelería que normalmente se incurren para el desarrollo de los procesos actuales en la unidad de sistemas tributarios de la división de informática de la DGI, se va a necesitar un material extra para la impresión del documento que la metodología estructura y además para los apuntes de las reuniones que se desarrollen según la planeación de cada proyecto. Los materiales necesarios se detallan en la siguiente tabla.

Tabla 27.
Precio de materiales de oficina (anexo 2).

Artículos	Cantidad Mensual	Precio Unitario (C\$)	Sub Total
Resma de papel BOND XEROX	4	121.47	485.88
Tóner de impresora HP	3	2,948.12	8844.36
Lapicero Papermate	30	4.54	136.2
Resaltador	10	18.67	186.7
Corrector	15	24.77	371.55
Caja de folder Ampo	3	145.46	436.38
Engrapadora	1	59.91	59.91
Caja de grapas	4	20.79	83.16
Fastener	3	35.36	106.08
Caja de clip	3	19.53	58.59
Perforadora	2	133.22	266.44
TOTAL, C\$			C\$ 11,035.25

Esto gastos serán mensuales permanentemente e inicialmente formarán parte de la inversión para la implementación de la metodología XP.

Gastos en capacitación

La capacitación del personal es indispensable para la implementación de la metodología de programación extrema en la gestión de proyectos de desarrollo de la unidad de sistemas tributarios, por lo que se hizo una cotización de un curso de metodología ágil de programación extrema.

El curso es de certificación Española con doble titulación y certificación internacional en más de 200 países, la modalidad es en línea, con un tiempo de duración de 120 horas, incluye material didáctico con envío sin costo adicional.

El costo de este curso es de **C\$ 16,560** por persona (**ver anexo 3**), el centro de estudios cuenta con un sistema de pagos flexible por cuotas, también cuenta con promociones permanentes según la temporada y cursos ofrecidos.

Debido a que la cantidad de empleados a capacitar de la unidad de sistemas tributarios es de 32 personas el costo total por capacitación es de **C\$ 529,920**.

Tabla 28.

Resumen de costos de capacitación (anexo 3).

Curso	Duración (días)	Costo por unidad(C\$)	Cantidad empleados	Total
Curso de Desarrollo de Metodologías Agiles	120	16,560	32	529,920

Costos totales de inversión.

Considerando los cuatro elementos previamente señalados: (adquisición de equipos, capacitación de la metodología, costo de alojamiento sistemas, gastos en papelería), se puede indicar el monto total aproximado de la inversión inicial a la que deberá incurrir la Dirección General de Ingresos por la implementación de la metodología de Programación Extrema en el proceso de desarrollo de sistemas.

Tabla 29.

Costos de Inversión Inicial por la implementación de la metodología XP.

DESCRIPCIÓN	COSTO (C\$)
Adquisición de equipos.	0
Capacitación de la metodología.	C\$ 529,920
Costo de alojamiento sistemas.	0
Gastos en papelería mensual	C\$ 11,035.25
Costo total	C\$ 540,955.25

Nota. Fuente: Elaboración Propia.

Beneficios

Los beneficios que se obtendrán por la implementación de la metodología de ágil de Programación Extrema son los siguientes:

- ✓ El proceso de desarrollo será más organizado y eficiente.
- ✓ Mayor productividad en el desarrollo de sistemas, debido a que es una metodología de desarrollo ágil.
- ✓ Mayor comunicación entre los integrantes del equipo de desarrollo.
- ✓ Cumplimiento de los tiempos planificados.
- ✓ Personal certificado con conocimiento de mejores prácticas en el desarrollo de sistemas.
- ✓ Satisfacción de usuarios de normativa y de los contribuyentes, ya que se haría un análisis detallado de la funcionalidad del proceso, reflejado en el producto terminado.

Conclusiones

Como resultados del desarrollo del presente trabajo de tesis, se realizó un diagnóstico de la situación actual en cuanto a la forma de trabajo, dando como resultados que el proceso de gestión de desarrollo de sistemas tiene deficiencias, que hacen que la planeación de desarrollo de sistemas sea variable y con estimaciones de tiempo inciertas, provocando retrasos de tiempo por solicitud de cambios en los sistemas una vez terminados por requerimientos mal entendidos en la fase de análisis.

Se realizó una propuesta del nuevo flujo de trabajo que implementa la metodología de desarrollo ágil de programación extrema, aplicando todas sus fases y herramientas que se deben seguir para que la División de Informática tenga un buen funcionamiento de un proceso estándar y organizado de desarrollo de sistemas, que le permita eliminar las deficiencias en el proceso, se analice mejor los requerimientos y optimicen el tiempo de desarrollo y dirijan esfuerzos a otros proyectos. Además se determinó que la metodología aplica al tipo de proceso que se desarrolla en la División de Informática y Sistemas.

Finalmente se estimaron los costos que se incurrirán por la implementación de la metodología de programación extrema en la unidad de sistemas tributarios dando como resultados un costo de inversión inicial de **C\$ 529,920**, que va a traer consigo beneficios que harán de la gestión de proyecto de desarrollo, un proceso maduro, organizado y eficiente.

Recomendaciones

- ✓ Se recomienda que la División de Informática y Sistemas implemente la metodología de programación extrema, ya que se determinó que aplica al tipo y naturaleza del proceso de desarrollo de sistemas, además se identificaron beneficios que harán que el proceso sea eficiente, maduro y organizado.

- ✓ Se recomienda seguir haciendo estudios que permitan la madurez en los procesos informáticos y prácticas de mejora continua.

Glosario

Automatizar: Aplicar procedimientos automáticos en la realización de un proceso o en una industria.

Bitácora: Es un documento que detalla una inconsistencia generada en un sistema en uso, esta se reporta al desarrollador encargado del mantenimiento de dicho sistema para que brinden la solución correspondiente.

COBIT: Es un marco de trabajo para auditar la gestión y control de los sistemas de información y tecnología, orientado a todos los sectores de una organización, es decir, administradores IT, usuarios y por supuesto, los auditores involucrados en el proceso.

Contribuyente: Contribuyente es toda persona física o jurídica que realiza una actividad económica o laboral que genera el pago de un impuesto y que asume una serie de deberes formales de acuerdo al Código Tributario y a las normas y resoluciones de la autoridad tributaria. Asimismo, son contribuyentes aquellas personas con bienes inmobiliarios gravados por la ley.

DGI: Iniciales de Dirección General de Ingresos.

DIS: Iniciales de División de Informática y Sistema.

Historias de Usuario: Las historias de usuario son descripciones de la funcionalidad de un proceso, que resumen la necesidad concreta de un usuario dentro de un proceso, así como la solución que la satisface.

Programación Extrema: es una metodología ágil y flexible utilizada para la gestión de proyectos. Esta metodología pone el énfasis en la retroalimentación continua entre cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes.

Requerimiento: Es un documento que detalla las funcionalidades de un proceso para realizar el desarrollo de un sistema o un módulo de uno existente en base a esas funcionalidades.

SCRUM: Es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa.

Seguridad Informática: La seguridad informática es una disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema informático. De todas formas, no existe ninguna técnica que permita asegurar la inviolabilidad de un sistema.

Sistema informático: Es un sistema que permite almacenar y procesar información; es el conjunto de partes interrelacionadas: hardware, software y personal informático.

UST: Iniciales de Unidad de Sistemas Tributarios.

Usuario: Es aquella persona que opera un sistema informático por medio de un dispositivo o un ordenador y realiza múltiples operaciones para lograr distintos propósitos u objetivos.

VET: Son las iniciales del sistema de Ventanilla Electrónica Tributaria. En este sistema los contribuyentes declaran sus impuestos.

XP: Son las iniciales de la metodología programación extrema.

Bibliografía

Bocciarelli & D'Ambrogio, (2011), modelar requisitos de seguridad en BP.

Booch G. & Rumbaugh J. (2011) UML el lenguaje unificado de modelado.

Cachero Castro, Cristina. (2008) OO-H: Una extensión de los métodos OO para el modelado y generación automática de interfaces hipermedia.

De acuerdo con J. William & C. Collazos (2009). Desarrollo basado en modelos de la interfaz de usuario de sistemas groupware.

Erljman Piwen, A., & Goyen Fros, A. (2001). Problemas y Soluciones en la Implementación de Extreme Programming. Montevideo.

Fernando Berzal, Francisco J. Cortijo & Juan Carlos Cubero. (s.f.). Desarrollo Profesional de Aplicaciones Web con ASP.NET. Recuperado el 14 de febrero de 2016, de <http://elvex.ugr.es/decsai/csharp/pdf/web/web-book-a4.pdf>

Ferrer, J. (2012). Conceptos generales de la arquitectura de aplicaciones web en implantaciones de aplicaciones web. Madrid, España: Ra-Ma Editorial, S.A.

Gauchat, J. D. (septiembre de 2013). El gran libro de HTML5, CSS3 y Javascript. Recuperado el 14 de febrero de 2016, de <https://adegiusti.files.wordpress.com/2013/09/el-gran-libro-de-html5-css3-y-javascript.pdf>

Gilfillán, I. (2008). La Biblia de MySQL. Recuperado el 14 de febrero de 2016, de <https://osmell.files.wordpress.com/2008/08/la-biblia-de-mysql-anaya-multimedia.pdf>

Heredero, C. d. (2008). Dirección y gestión de los sistemas de información en la empresa. En C. d. Heredero. Madrid.

Hernandez Sampieri, Fernandez Collado, Baptista Lucío. (2009). Metodología de la Investigación.

Investigacion holistica. (febrero 2008). Recuperado el 20 de abril de 2016, de <http://investigacionholistica.blogspot.com/2008/02/la-investigacin-proyectiva.html>

Jorge Ferrer, Víctor García, Rodrigo García. (s.f.). CURSO COMPLETO HTML. Recuperado el 14 de febrero de 2016, de <http://es.tldp.org/Manuales-LuCAS/doc-curso-html/doc-curso-html.pdf>

Joskowicz, J. (2008). Reglas y Prácticas en Extreme Programming. Recuperado el 2015, de <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>

La propuesta de Ingeniería Web basada en UML (2012). Recuperado el 14 de febrero de 2016, <http://www.slideshare.net/coesiconsultoria/4-desarrollo-gil-del-software> <http://maestria-modulo7.blogspot.com/2012/04/procesos-de-desarrollo-ligeros>.

Laudon, K. C., & Laudon, J. P. (2012). Sistemas de Información Gerencial. México.

López, P y Francisco, R. s/f. INGENIERÍA DEL SOFTWARE. (En línea). VE. Consultado, 6 de Mayo de 2015. Formato PDF. Disponible en: <http://ocw.unican.es/enseñanzas-tecnicas/ingenieria-del-software-i/materiales-de-clase-1/is1-t02-trans.pdf>

Mora, J. T. (agosto de 2011). Arquitectura de software para aplicaciones web. Recuperado el 27 de marzo de 2016, de Juan Tahuiton Mora.

NEWCOMLAB. (2013). Recuperado el 14 de febrero de 2016, de http://www.newcomlab.com/default.aspx?id_seccion=936

Pacheco, I y García, J. 2008. Una Metodología Basada en Prácticas Efectivas para Desarrollar Software Educativo.MX. Postgraduate Department, Technological University of the Mixtec Region. vol.11 no.4 . (En línea). Consultado, 6 Mayo de 2015. Formato PDF. Disponible en: http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S0798-40652010000400003&lang=pt

Pérez, J. E. (s.f.). Introducción a JavaScript.

Pressman (2015). Software Engineering

Pressman, R. 2010. Ingeniería del Software Un Enfoque Práctico. 7ma ed. University ofConnecticut. McGraw-Hill Interamericana Editores, S.A.

Programación Luis Moquete. (12 de junio de 2013). Recuperado el 14 de febrero de 2016, de <http://programacionluismoquete.blogspot.com/2013/06/el-iss-significa-internet-information.html>

RN Taylor, N. Medvidovic, EM Dashofy . (2009). Software Architecture: Foundations, Theory, and Practice.

Rodríguez, A. (2009). Metodología de Diseño usada en Ingeniería. México.

Rodríguez, E. 2012. Conceptos básicos de Ingeniería de Software. (En línea). VE. Consultado, 6 de Mayo de 2015. Formato PDF. Disponible en: <http://www.tamps.cinvestav.mx/~ertello/swe/sesion01.pdf>

Seco, J. A. (s.f.). Lenguaje de Programación C#. Recuperado el 14 de Febrero de 2016, de <http://dis.um.es/~bmoros/privado/bibliografia/LibroCsharp.pdf>.

Schwabe, G. Rossi (2008) Desarrollo de aplicaciones hipermedia con OOHDM.

Universidad Nacional de la Plata (2009). Argentina. Metodologías de diseño usadas en Ingeniería Web, su vinculación con las NTICS.

Wordpress. (25 de enero de 2013). Recuperado el 27 de marzo de 2016, de <https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>

Zuniga. (s.f.). UV. Recuperado el 14 de Febrero de 2016, de http://www.uv.es/zuniga/06_La_distribucion_normal_o_de_Gauss.pdf

Anexos

Anexo 1. Entrevistas

Entrevista

Permítame saludarle muy cordialmente, esta entrevista está dirigida a usted, Director de Informática de la Dirección General de Ingresos con el objetivo de conocer su opinión y un poco de sus buenos oficios relacionado a la gestión de proyectos de desarrollo de sistemas. Información que servirá para un trabajo de investigación, por lo que se requiere su colaboración al responder las siguientes preguntas:

1. ¿Cómo considera usted la gestión de los proyectos de desarrollo en la Unidad de Sistemas Tributarios de la División de Informática?
2. ¿Qué tan definido cree usted que es el proceso de gestión de los proyectos de desarrollo de sistemas?
3. ¿Considera que se cumplen los tiempos de desarrollo de forma eficiente?
4. ¿Cuáles son sus funciones relacionado a la gestión de los proyectos de desarrollo de sistemas?
5. ¿Existe manual operativo o una metodología de gestión del desarrollo de sistemas, en caso de que exista, cuál es?
6. ¿Cuáles cree usted que son las debilidades o mayores problemas en el proceso de gestión de proyectos de desarrollo de sistemas?
7. ¿Considera que se puede mejorar el proceso de gestión de los proyectos de desarrollo de sistemas en la Unidad de Sistemas Tributarios?, en que aspectos.
8. ¿Estaría de acuerdo que se implemente una metodología de desarrollo ágil para gestionar los proyectos de desarrollo de sistemas?
9. ¿Está dentro de sus funciones una toma de decisión de implementar una metodología de desarrollo ágil para gestionar los proyectos de desarrollo de sistemas? ¿De quién depende?

Entrevista

Permítame saludarle muy cordialmente, esta entrevista está dirigida a usted, Jefe de Oficina de Sistemas de la División de Informática de la Dirección General de Ingresos con el objetivo de conocer su opinión y un poco de sus buenos oficios relacionado a la gestión de proyectos de desarrollo de sistemas. Información que servirá para un trabajo de investigación, por lo que se requiere su colaboración al responder las siguientes preguntas:

1. ¿Cómo considera usted la gestión de los proyectos de desarrollo en la Unidad de Sistemas Tributarios de la División de Informática?
2. ¿Qué tan definido cree usted que esta el proceso de gestión de los proyectos de desarrollo de sistemas?
3. ¿Considera que se cumplen los tiempos de desarrollo de forma eficiente?
4. ¿Considera que el factor tiempo en algunos casos genera vacíos en el entendimiento de los requerimientos?
5. ¿Cuáles son sus funciones relacionado a la gestión de los proyectos de desarrollo de sistemas?
6. ¿Existe manual operativo o una metodología de gestión del desarrollo de sistemas, en caso de que exista, cuál es?
7. ¿Cuáles cree usted que son las debilidades o mayores problemas en el proceso de gestión de proyectos de desarrollo de sistemas?
8. ¿Considera que se puede mejorar el proceso de gestión de los proyectos de desarrollo de sistemas en la Unidad de Sistemas Tributarios?, en que aspectos.
9. ¿Estaría de acuerdo que se implemente una metodología de desarrollo ágil para gestionar los proyectos de desarrollo de sistemas?

Entrevista

Permítame saludarle muy cordialmente, esta entrevista está dirigida a usted, Jefe de Unidad de Sistemas Tributarios de la División de Informática de la Dirección General de Ingresos con el objetivo de conocer su opinión y un poco de sus buenos oficios relacionado a la gestión de proyectos de desarrollo de sistemas. Información que servirá para un trabajo de investigación, por lo que se requiere su colaboración al responder las siguientes preguntas:

1. ¿Cómo considera usted la gestión de los proyectos de desarrollo en la Unidad de Sistemas Tributarios de la División de Informática?
2. ¿Qué tan definido cree usted que esta el proceso de gestión de los proyectos de desarrollo de sistemas?
3. ¿Considera que se cumplen los tiempos de desarrollo de forma eficiente?
4. ¿Considera que el factor tiempo en algunos casos genera vacíos en el entendimiento de los requerimientos?
5. ¿Cuáles son sus funciones relacionado a la gestión de los proyectos de desarrollo de sistemas?
6. ¿Cuáles son los objetivos de la Unidad de Sistemas Tributarios?
7. ¿Existe manual operativo o una metodología de gestión del desarrollo de sistemas, en caso de que exista, cuál es?
8. ¿Cuáles cree usted que son las debilidades o mayores problemas en el proceso de gestión de proyectos de desarrollo de sistemas?
9. ¿Considera que se puede mejorar el proceso de gestión de los proyectos de desarrollo de sistemas en la Unidad de Sistemas Tributarios?, en que aspectos.
10. ¿Estaría de acuerdo que se implemente una metodología de desarrollo ágil para gestionar los proyectos de desarrollo de sistemas?

Entrevista

Permítame saludarle muy cordialmente, esta entrevista está dirigida a usted, Analista de sistemas de la Unidad de Sistemas Tributarios de la División de Informática, con el objetivo de conocer su opinión y un poco de sus buenos oficios relacionado a la gestión de proyectos de desarrollo de sistemas. Información que servirá para un trabajo de investigación, por lo que se requiere su colaboración al responder las siguientes preguntas:

1. ¿Cómo considera usted la gestión de los proyectos de desarrollo en la Unidad de Sistemas Tributarios de la División de Informática?
2. ¿Qué tan definido cree usted que esta el proceso de gestión de los proyectos de desarrollo de sistemas?
3. ¿Considera que se cumplen los tiempos de desarrollo de forma eficiente?
4. ¿Cuáles son sus funciones relacionado a la gestión de los proyectos de desarrollo de sistemas?
5. ¿Existe manual operativo o una metodología de gestión del desarrollo de sistemas, en caso de que exista, cuál es?
6. ¿Cuáles cree usted que son las debilidades o mayores problemas en el proceso de gestión de proyectos de desarrollo de sistemas?
7. ¿Considera que se puede mejorar el proceso de gestión de los proyectos de desarrollo de sistemas en la Unidad de Sistemas Tributarios?, en que aspectos.
8. ¿Estaría de acuerdo que se implemente una metodología de desarrollo ágil para gestionar los proyectos de desarrollo de sistemas?

Anexo 2. Cotización Gonper librerías.

GONPER LIBRERIAS GONZALEZ PEREIRA S.A. Sucursal METROCENTRO, RUC No. 30310000001499 Segundo nivel, contiguo a Cinecari Teléfonos 2226-3610/ 2226-3609		COTIZACION						
Usuario: METRO	Fecha Emisión: 14/02/2019	Vencimiento: 29/02/2019	Hora: 6:39:31 pm	Precio:P1				
Almacén: MC	Cliente: 061	RUC:	Vendedor: CLAUDIA MORALES	Cotiz. No: H-000030110				
Dirección:			Membresía: NO TIENE MEMBRESIA	Desc: 0.00%				
Cant	Descripción	Código	Uapuesto	Udesc	P.UNIT	Ejeto	CPVAGG	
1	PAPEL BOND 40 TO XEROX 8.5x11 ULTRA BLANCO 75GR	7897973800026	15.00	1	0.00	121.47	121.47	
1	FOLDER ANPD MANILA T/CARTA CAJA 100UDS	7441046302001	15.00	1	0.00	145.46	145.46	
12	LAPICERO PAPERMATE KILOMETRICO 190 ST AZUL 1.0 MM	3401170414217	0.00	1	0.00	4.54	54.48	
1	RESALTADOR SHARPIE SURTIDO 25033	855URT	0.00	1	0.00	18.67	18.67	
1	CORRECTOR LIQUID PAPER T/BOLIGRAF0 7ML 50382-L661321580	8854536000324	0.00	1	0.00	24.77	24.77	
1	CORRECTOR PAPERMATE LIQUID PAPER CINTA E/BLISTER 8.5M	041540066040	0.00	1	0.00	78.60	78.60	
1	SOBRE PLASTICO ACME/BARRILITO T/L HORIZONTAL 120FCH	7501214901761	15.00	1	0.00	22.32	22.32	
1	ENGRAPADORA ACME MEDIA TIRA 40M TOP105	7501214901419	15.00	1	0.00	59.91	59.91	
1	GRAPAS BARRILITO STANDARD 26/6 CJ-5000UD 101068A	7501214979449	15.00	1	0.00	20.79	20.79	
1	FASTENER ACME/BARRILITO 80MM CJ-50 UDS 946	7501214999423	15.00	1	0.00	35.36	35.36	
1	CLIPS ACME/BARRILITO NIQUELADOS CLP50MM CJ-100UD	7501214904557	15.00	1	0.00	19.53	19.53	
1	PERFORADORA ACME 8501 MEDIANA 20H 2 HUECOS	7501214901181	15.00	1	0.00	133.22	133.22	
1	TONER HP 07115A HEWLETT PACKARD	723184918454	15.00	1	0.00	146.12	146.12	
(ESTAMOS EXENTOS DE RETENCIONES DE I.R.)								
Comentario: GERALD GONZALES ZELEDON 5702-8362 Teléfono: 5702-8362						Sub Total:	176.52	3,506.18
						Descuentos:	0.00	0.00
						Totales:		3,682.78
						Retenciones:		0.00
						Más IVA:		325.93
						Gran Total:		C\$ 4,208.63
COTIZACION		ESTE DOCUMENTO NO SUSTITUYE NINGUNA FACTURA						

Figura 10. Cotización Gonper.

Anexo 3. Costo curso de metodologías ágiles

The screenshot displays the Euroinnova Business School website for the course 'Metodologías Ágiles'. The page features a navigation bar, a course title, a star rating, and a promotional banner. A 'RESUMEN DE LA COMPRA' table is visible on the right side of the page.

RESUMEN DE LA COMPRA	
Precio Original	33120 NIO
Ahorro	16560 NIO
Precio Final:	16560 NIO

Additional details from the page include: 'Metodologías Ágiles. Realiza este Curso ONLINE y HOMOLOGADO de Metodologías Ágiles. Hazte experto en el Desarrollo de Metodologías Ágiles, gracias a este Curso con Doble Titulación.' and a payment summary showing '4140NIO / mes' with a '0% InnovaPAY' logo.

Figura 11. Cotización de curso en Euroinnova.

Fuente: https://www.euroinnova.edu.es/Desarrollo-Metodologias-Agiles?fbclid=IwAR1HFEs7_7HWxVIKXgm-Vz3OOA46YDkN2ZMnMF_4uNEuKMV6Y186DL2aRGw#info-complementaria