

Universidad de las Ciencias Informáticas “Facultad 3”



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

Título: Herramienta para evaluar la satisfacción de los clientes internos en el proceso de desarrollo de software de los proyectos del Centro de Gobierno Electrónico de la Facultad 3.

Autor: Yaima Morales Labrada.

Tutor: Ing. Raúl Velázquez Alvarez.

Co-Tutor: Ing. Reinier Silverio Figueroa.



“Nos acompaña la convicción más profunda de que las ideas pueden más que las armas por sofisticadas y poderosas que estas sean”



Declaración de Autoría

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Yaima Morales Labrada

Ing. Raúl Velázquez Alvarez



Datos de Contacto

Datos de contacto

Tutor: Raúl Velázquez Álvarez. Ingeniero en Ciencias Informáticas, Profesor Instructor, Jefe del Grupo de Calidad, Asesor de la Calidad del Software del Centro CEGEL, 4 años de experiencia, correo electrónico: rvelazquez@uci.cu.

Co-Tutor: Reinier Silverio Figueroa. Ingeniero en Ciencias Informáticas, Arquitecto de software del Centro CEGEL, correo electrónico: rsilveriof@uci.cu.



Agradecimientos

AGRADECIMIENTOS

A mi mamá, por haber confiado en mí durante todos estos años, por apoyarme en todas mis decisiones, por su comprensión, por su preocupación en los momentos difíciles, por todo su amor y porque sin ella hoy no estaría aquí.

A mi papá, por estar siempre pendiente de mis estudios, por inculcarme valores para ser una mejor persona, por darme su amor sincero y por creer en mí aún cuando yo no lo hacía.

A mi tía Nerita, por ser una segunda madre para mí, por todos sus consejos, por su ayuda siempre que la necesité y por exigirme para que obtuviera los mejores resultados como estudiante.

A mi abuela Neris, por el amor que siempre me ha dado, por su ternura, por preocuparse por mí y por estar a mi lado en todo momento.

A mi hermana, por todo su cariño y apoyo.

A Rodrigo, por quererme como a una hija y por haberme brindado su apoyo en los momentos buenos y en los malos a lo largo de la carrera.

A mi novio, por darme fuerzas y amor para continuar, por haber estado a mi lado a pesar de la distancia y por confiar en mí hasta el último momento.

A Raque, por todo su cariño y su ayuda durante todo este período.

A toda mi familia, por ser tan especial, por quererme con mis virtudes y defectos y por apoyarme siempre.

A Ariagna, por el cariño que me brindó, por preocuparse por mí, por sus buenos consejos y por ser mi familia cuando no tenía la mía cerca.

A Dibeikys, a Marieta y a Yunaily por su apoyo y por el cariño que me han dado desde que me conocieron.

A mi tutor y a mi co-tutor, por su paciencia, por brindarme su ayuda en los momentos difíciles y por exigirme al máximo para que todo saliera bien.

A Martí, por su cariño y por haberme ayudado en todo lo que necesité durante los 5 años de la carrera.

A mi amiga Marta, por haberme permitido compartir con ella todos estos años de estudios.

A mis compañeros del grupo y del proyecto, por los buenos momentos que pasé junto a ellos.

A mis profesores, por guiarme y por formar en mí una profesional.

En general, a todas aquellas personas que de una forma u otra me ayudaron con la realización de esta investigación.

A todos, muchas gracias.



Dedicatoria

DEDICATORIA

Quiero dedicarle mi trabajo de diploma a mi mamá, por su esfuerzo y sacrificio para que pudiera terminar la carrera, por haber sido mi motor impulsor, por ser mi amiga, por tenerme confianza, por haber soñado tanto con este momento y porque hoy estoy aquí gracias a ella.

A mi papá, por su amor infinito, por guiarme ante la vida y por formar en mí una mujer de bien.

A mi tía Nerita, porque siempre ha estado ahí cuando la he necesitado, por ser un ejemplo para mí, por todas sus enseñanzas y por quererme como a una hija.

A mi abuela Neris, por brindarme todo su amor, por preocuparse por mis estudios y por haber estado a mi lado durante toda la vida dándome siempre su cariño y su apoyo.

A mi hermana, porque me ha brindado su apoyo a lo largo de la carrera.

A mis sobrinas, que son mi mayor tesoro, por todas las alegrías que me han dado.

A Rodrigo, por ayudarme durante todo este período y por darme su amor incondicional.

A mi novio, por la felicidad que me ha dado y por su apoyo en todo momento.

A Raque, por ayudarme siempre y por quererme como a una sobrina.

A toda mi familia, porque siempre se han preocupado por mí y me han dado su cariño más sincero.



Índice

CONTENIDO

DECLARACIÓN DE AUTORÍA.....	III
Agradecimientos	V
Dedicatoria.....	VI
Resumen.....	XI
Introducción	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Calidad de Software	6
Calidad total.....	7
1.2 Satisfacción del cliente.....	8
Importancia de medir la satisfacción del cliente interno	10
Evaluación de la satisfacción del cliente interno en el mundo y en Cuba.....	10
Sistemas informáticos que miden la satisfacción del cliente interno.....	13
1.3 Selección de la metodología de desarrollo.....	14
Programación Extrema (XP)	14
Proceso Unificado de Desarrollo (RUP).....	14
1.4 Selección de herramientas, tecnologías, lenguajes, arquitectura y tipo de aplicación informática.....	15
Herramientas de modelado.....	15
❖ Rational Rose.....	16
❖ Visual Paradigm.....	16
Herramientas de desarrollo.....	17
❖ NetBeans 7.3	17

Índice

❖ Apache 2.2	17
❖ LimeSurvey 2.0	17
Lenguajes	19
❖ PHP 5.3	19
❖ HTML 5	19
❖ CSS 3.....	20
❖ Javascript 1.8.5.....	20
❖ UML 2.1 (Lenguaje Unificado de Modelado).....	20
Arquitectura Modelo-Vista-Controlador (MVC).....	20
1.5 Sistema Gestor de Base de Datos	21
Sistema Gestor de Base de Datos MySQL.....	22
1.6 Técnicas, métricas y pruebas a emplear para la verificación y la validación del sistema.....	22
Técnicas de validación de requisitos.....	22
Revisiones de requisitos:.....	23
Construcción de prototipos:.....	23
Métricas.....	23
Métricas para Requisitos	23
Métricas para diseño.....	23
Pruebas.....	23
Pruebas unitarias.....	24
Pruebas de aceptación	24
1.7 Conclusiones del capítulo	25
CAPÍTULO 2: Análisis, diseño e implementación de la propuesta de solución	26

Índice

2.1 Definición de dimensiones, roles, atributos y método de medición para evaluar la satisfacción de los clientes internos	26
2.2 Descripción del sistema	31
Escenario actual del proceso de evaluación de la satisfacción del cliente interno en el Centro	31
Propuesta del sistema	31
Requisitos funcionales	32
Requisitos no funcionales.....	32
Historias de usuario	35
2.3 Planificación	36
Estimación de esfuerzo por Historias de Usuario.....	37
Plan de iteraciones.....	38
Plan de duración de las iteraciones.....	38
Plan de entregas	39
2.4 Diseño.....	40
Patrón de Arquitectura.....	40
Patrón MVC	40
Patrones de Diseño.....	43
Tarjetas CRC (Clase – Responsabilidad – Colaborador).....	45
Diseño de la base de datos	46
2.5 Codificación.....	47
Estándares de codificación.....	47
Tareas de Ingeniería	49
2.6 Conclusiones del capítulo	50

Índice

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN	51
3.1 Aplicación de técnicas de validación de requisitos y utilización de métricas	51
Técnicas de validación de requisitos	51
Métrica para requisitos	52
Métricas para diseño	54
3.2 Verificación del sistema	59
Pruebas unitarias	59
❖ Pruebas de caja negra	59
❖ Pruebas de caja blanca	60
3.3 Validación de sistema	63
Pruebas de aceptación	63
3.4 Validación de las variables	64
3.5 Conclusiones parciales	66
Conclusiones generales	67
Recomendaciones	67
Referencias bibliográficas	68



RESUMEN

En la actualidad, muchas empresas se dedican a la producción de software y su principal objetivo es desarrollar productos de calidad, para lograr el éxito en este sentido no se puede obviar la satisfacción de los clientes, no solo es complacer al usuario final, también se debe prestar especial atención a las personas que llevan a cabo el proceso de desarrollo de software pues el resultado de su trabajo es lo que determinará que el producto terminado sea del agrado del cliente.

El presente trabajo está dedicado a la realización de una herramienta para evaluar la satisfacción de los clientes internos en el proceso de desarrollo de software de los proyectos del Centro de Gobierno Electrónico de la Facultad 3, para ello se realiza un estudio del estado del arte de la temática a tratar, identificando los conceptos principales de la investigación referentes a calidad de software, calidad total y satisfacción del cliente. Además, se hace un análisis de los sistemas informáticos que miden la satisfacción del cliente a nivel mundial, demostrándose que estos no resuelven los problemas del Centro. Para guiar el desarrollo del sistema a implementar se escogió como metodología de desarrollo de software XP, utilizando como lenguaje de programación PHP y empleando el IDE de desarrollo Netbeans. Se obtuvo como resultado una herramienta que mide el Índice de Satisfacción del Cliente Interno mediante la realización de encuestas y que permite identificar las causas de insatisfacción de los clientes internos del Centro.

Palabras clave:

Calidad de software, cliente interno, Índice de Satisfacción del Cliente Interno, satisfacción del cliente.



Introducción

INTRODUCCIÓN

En la actualidad, la humanidad ha adquirido un considerable desarrollo tecnológico a gran escala debido al avance de las Tecnologías de la Información y las Comunicaciones (TIC), logrando impulsar de manera creciente la evolución en innumerables sectores científicos. Es indudable que la informática se ha convertido en la ciencia clave para el progreso en casi todas las esferas de un país y en el transcurso de los últimos años la producción de software ha ocupado un lugar significativo en el ámbito económico mundial.

Anteriormente, muchas de las empresas productoras de software centraban su desarrollo en la creación de un producto determinado, hoy en día estas empresas no solo se dedican a la producción, también se enfocan en brindar la atención que merecen los clientes, tratando siempre de satisfacer sus necesidades. Lograr la satisfacción del cliente constituye un requisito indispensable que toda empresa debe tener en cuenta para alcanzar los objetivos propuestos. Satisfacer al usuario final es una tarea primordial de los Centros de Desarrollo de Software, pero en muchas ocasiones no se tiene en cuenta al cliente interno (personal que lleva a cabo el desarrollo del software), sin embargo no cabe duda que el tener clientes internos satisfechos constituye uno de los factores clave para alcanzar el éxito en los negocios porque el esfuerzo realizado por el equipo de desarrollo contribuye de forma directa al logro de un producto final con calidad.

Cuba, independientemente de no encontrarse entre los países desarrollados incorpora el uso de las TIC a la vida cotidiana del pueblo. Se propone llevar a la sociedad cubana la revolución informática. La Universidad de las Ciencias Informáticas (UCI) juega un papel muy importante en todo el proceso de informatización que se lleva a cabo en el país, cuenta con tecnología informática avanzada, con la cual se controla la información de todos los estudiantes y profesores vinculados al centro, tanto en la docencia como en la producción.

En la UCI, el Centro de Gobierno Electrónico (CEGEL) de la Facultad 3, tiene como misión satisfacer las necesidades de clientes gubernamentales mediante el desarrollo de productos, servicios y soluciones integrales de alta confiabilidad, calidad, competitividad, fidelidad y eficiencia, a partir de un personal altamente calificado.

Introducción

El Centro CEGEL se preocupa por el estado de satisfacción de sus clientes internos, para llevar a cabo el proceso de evaluación utiliza como técnica la aplicación de encuestas, este proceso está sujeto a las siguientes condiciones:

- Se cuenta con encuestas diseñadas para cada rol. En el Centro hay carencias de recursos de impresión que cubran las necesidades de los clientes internos de todos los proyectos de la facultad, lo que trae como consecuencia que sea un proceso costoso imprimir las encuestas para medir la satisfacción del cliente interno.
- La realización manual de este proceso requiere esfuerzo personal a la hora de buscar la información para obtener los resultados de la encuesta.
- Existe una deficiente digitalización de los datos que provoca la pérdida de información.
- La ineficiencia en la inmediatez de los resultados afecta el proceso de toma de decisiones administrativas en el Centro.

Cada uno de los elementos antes mencionados, permiten observar que las deficiencias en el proceso de recolección y análisis de los datos, dificultan la identificación de causas de insatisfacción como apoyo a la toma de decisiones administrativas en el Centro CEGEL.

Esta situación de conflicto da lugar al siguiente problema de investigación: ¿Cómo contribuir a la mejora del proceso de evaluación de la satisfacción del cliente interno en el proceso de desarrollo de software de los proyectos del Centro CEGEL de manera que se facilite la identificación de causas de insatisfacción como apoyo a la toma de decisiones administrativas en dicho Centro?

Siendo el **objeto de estudio** el proceso de evaluación de la satisfacción del cliente.

El **objetivo general** de la investigación es: desarrollar una herramienta para evaluar la satisfacción del cliente interno en el proceso de desarrollo de software en los proyectos del Centro CEGEL que facilite la identificación de causas de insatisfacción como apoyo a la toma de decisiones administrativas en dicho Centro.

Introducción

Y como **campo de acción** proceso de evaluación de la satisfacción del cliente interno en proyectos de desarrollo de software.

Se plantea para este trabajo la siguiente **idea a defender**: con el desarrollo de una herramienta para evaluar la satisfacción del cliente interno en el proceso de desarrollo de software de los proyectos del Centro CEGEL se facilita la identificación de causas de insatisfacción como apoyo a la toma de decisiones administrativas en dicho Centro.

El objetivo general se desglosa en una serie de **objetivos específicos**:

- ✓ Elaborar el marco teórico de la investigación para definir el estado del arte del proceso de evaluación de la satisfacción del cliente interno.
- ✓ Desarrollar la propuesta de solución para dar respuesta a la problemática existente.
- ✓ Validar la solución propuesta para comprobar que la misma responde a las necesidades del cliente.

Se plantean las siguientes **tareas de investigación** para dar cumplimiento a los objetivos:

- ✓ Realización de un estudio del estado del arte de la temática en la UCI, Cuba y el mundo para definir los aspectos más significativos a tener en cuenta para el desarrollo del trabajo.
- ✓ Análisis de las metodologías de desarrollo, herramientas y tecnologías posibles a utilizar para el desarrollo de la solución.
- ✓ Levantamiento de requisitos para precisar las exigencias de la herramienta que desea el cliente.
- ✓ Análisis, diseño e implementación del sistema para obtener la aplicación requerida por el cliente final.
- ✓ Realización de las pruebas de verificación y validación a la aplicación para comprobar que la misma funciona correctamente y que se cumplen los requisitos definidos por el cliente.

Métodos teóricos.

- **Método analítico - sintético:**

A través de un estudio profundo de la teoría y las tendencias existentes en el mundo del proceso de evaluación de la satisfacción del cliente se sintetizan los aspectos más relevantes a considerar para el desarrollo del trabajo.

- **Método análisis histórico – lógico:**

Se adquiere la información relacionada con la evolución que ha tenido el proceso de evaluación de la satisfacción del cliente, además, se investiga acerca de los sistemas informáticos que se utilizan para desarrollar dicho proceso.

- **Modelación:**

Se representa el proceso que se lleva a cabo para evaluar la satisfacción de los clientes internos a través de las Historias de Usuario describiendo el comportamiento del sistema, además se explican las Tareas de Ingeniería y se confeccionan las tarjetas Clase Responsabilidad-Colaborador que contienen la información de las clases.

Métodos empíricos.

- **Entrevista:**

Se recopila información acerca de los métodos que son utilizados en los proyectos del Centro para medir la satisfacción de los clientes internos y la forma en que se realiza este proceso actualmente.

- **Medición:**

Se utilizan métricas para la validación del diseño y de los requisitos del sistema, así como las pruebas para verificar el correcto funcionamiento de la aplicación.

La **estructura de la investigación** es la siguiente:

Consta de introducción, 3 capítulos, conclusiones, recomendaciones y referencias bibliográficas

Capítulo #1. Fundamentación teórica: se realiza un estudio del estado del arte que permita conocer la situación mundial del tema, además se hace referencia a los principales conceptos referentes al mismo. Se describen los sistemas informáticos que existen para evaluar la satisfacción de los clientes, por último se caracterizan las metodologías de desarrollo, tecnologías y herramientas posibles a utilizar, seleccionando las que más se ajustan a las necesidades del sistema a implementar, con el fin de desarrollar una aplicación web que permita identificar las causas de insatisfacción de los clientes internos del Centro CEGEL de la Facultad 3.

Capítulo #2. Análisis, diseño e implementación de la propuesta de solución: en este capítulo se determinan los requisitos funcionales y no funcionales que la herramienta a implementar debe cumplir, además se presenta la descripción del sistema y se determinan las personas relacionadas con la aplicación. También se realiza la descripción del comportamiento del sistema a través de las Historias de

Introducción

Usuario, se establece un tiempo estimado en semanas para implementar cada Historia de Usuario , luego se confecciona el Plan de iteraciones, se define el tiempo de duración que tendrá cada una y se elabora el Plan de entregas. Como parte del diseño se define el patrón de arquitectura y se identifican los patrones de diseño que se van a utilizar, posteriormente se procede a confeccionar las Tarjetas CRC (Clase-Responsabilidad-Colaborador) que contienen el nombre de las clases, las responsabilidades y sus colaboradores. Luego se diseña la base de datos, se elaboran las Tareas de Ingeniería y se determinan los estándares de codificación para llevar a cabo la implementación del sistema.

Capítulo #3: Validación de la propuesta de solución: en este último capítulo se aplican métricas para requisitos y para diseño, se utilizan las pruebas unitarias para verificar que el software cumple los requisitos funcionales y no funcionales de su especificación, para ello se utilizan los métodos de prueba de caja blanca y de caja negra, además se emplean las pruebas de aceptación para confirmar si las historias han sido implementadas correctamente. Por último, se realiza la validación de las variables para constatar que se cumplió el problema de la investigación planteado.

Capítulo 1: Fundamentación Teórica

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

La satisfacción del cliente es un elemento fundamental para toda empresa que desea tener éxito y obtenerla es uno de los resultados más importantes de prestar servicios de buena calidad. Complacer al cliente garantiza numerosos beneficios para una empresa y para lograrlo es indispensable realizar periódicamente una evaluación de la satisfacción para así conocer las expectativas que los clientes esperan alcanzar.

En este capítulo se realiza un estudio de las definiciones de los términos fundamentales que se usarán a lo largo de la investigación, además se explican las herramientas, tecnologías y metodologías más utilizadas para el desarrollo de una aplicación web y de esta forma seleccionar las que más se ajusten a las necesidades del sistema a implementar.

1.1 Calidad de Software

En la actualidad, a nivel mundial la industria del software se ha convertido en un enorme gigante, el cual crece y se desarrolla a ritmo creciente. Todas las empresas quieren producir aplicaciones informáticas con alta calidad, en el menor tiempo posible y a costos mínimos. La calidad del software juega un papel importante hoy en día, influyendo positivamente en la decisión de un cliente.

La IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) define la calidad de software como *“el grado con el que un sistema, componente o proceso cumple con los requisitos especificados y las necesidades o expectativas del cliente o usuario.”* ((IEEE), 1990)

Según Roger Pressman, *“la calidad de software es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente.”* (Pressman, 1998)

A partir de los conceptos anteriormente referenciados, se define la calidad de software como el cumplimiento de los requisitos establecidos por un cliente para la creación de un producto de software determinado.

Con el avance de la industria del software se ha producido una prolongada crisis del software, donde los productos se entregan con demoras, los desarrollos exceden lo inicialmente presupuestado y no se cumple con los requisitos originales, por ello a medida que progresa, la calidad total se vuelve más que un sello de eficiencia y eficacia, una necesidad.

Calidad total

“La Calidad Total es una estrategia de gestión de la organización que tiene como objetivo satisfacer de una manera equilibrada las necesidades y expectativas de todos sus grupos de interés (en general, los clientes, empleados y la sociedad en general)”. (Fundación Vasca para la Excelencia, 2011)

Para Ishikawa, *“la Calidad Total es cuando se logra un producto económico, útil y satisfactorio para el consumidor”.* (Vivó, 2012)

Según Crosby, *“la Calidad Total es el cumplimiento de los requerimientos, donde el sistema es la prevención, el estándar es cero defectos y la medida es el precio del incumplimiento”.* (Vivó, 2012)

La Calidad Total es un proceso de mejora continua y es por tanto una actitud de tratar de hacer mejor las cosas. Para que funcione, debe ser entendido por toda la organización, de esta manera se enfoca el proceso de mejora desde la resolución de problemas, hasta aumentar la comunicación entre todos los empleados de la empresa.

En el contexto de la Calidad Total, el componente más importante de una organización son las personas. Sólo a partir de la inteligencia, de la voluntad y de la iniciativa de las personas, las tareas son hechas en función de la mejora continua de la calidad. Por esto, la forma de trabajo que impera en las organizaciones con Calidad Total es el trabajo en equipo, para encarar y resolver los problemas que van presentándose.

Hoy en día, la Calidad Total es el resumen de las mejores prácticas en el ámbito de la gestión de organizaciones. A estas mejores prácticas, se les suele denominar los ocho Principios de la Calidad Total:

1. Orientación hacia los resultados.
2. Orientación al cliente.
3. Liderazgo y coherencia en los objetivos.
4. Gestión por procesos y hechos.

5. Desarrollo e implicación de las personas.
6. Aprendizaje, innovación y mejora continuos.
7. Desarrollo de alianzas.
8. Responsabilidad social. (Fundación Vasca para la Excelencia, 2011)

Es preciso aplicar cada una de estas prácticas en las organizaciones para cumplir con las expectativas del cliente, de este modo se obtendrán mejores resultados y ocurrirá un profundo cambio en la manera de concebir y manejar la organización, donde el progreso constante de todos los procesos es fundamental, por lo tanto se concluye que la Calidad Total en el área de atención al cliente, comienza comprendiendo los gustos, las necesidades y las perspectivas del cliente para luego satisfacerlas.

Es necesario que toda organización que decida transitar por el camino de la Calidad Total esté convencida de que realiza esta inversión para asegurar su futuro y el de su gente y se proponga orientar los esfuerzos de todo el personal, para mejorar día a día la calidad, la productividad y la competitividad y así poder ofrecer la máxima satisfacción a sus clientes, e incluso, superar sus expectativas.

1.2 Satisfacción del cliente.

La satisfacción del cliente es la medida perceptual de lo que la empresa hace. La innovación es un signo de los tiempos actuales, la velocidad con que las cosas cambian, se reinventan, se copian y vuelven a cambiar es impresionante. Superar las expectativas del cliente es un reto que no termina jamás, es la esencia de la mejora continua.

ISO 9001(Organización Internacional para la Estandarización) impulsa a las organizaciones a que alcancen la satisfacción del cliente. Sitúa este objetivo en su punto de mira y aboga por la mejora del sistema de gestión de la calidad como vehículo hacia su consecución. En ISO 9001 la meta no es la calidad, es aumentar la satisfacción del cliente. (La satisfacción del cliente en ISO 9001, 2008)

Pereiro plantea lo siguiente: *“La satisfacción de las necesidades y expectativas del cliente constituye el elemento más importante de la gestión de la calidad y la base del éxito de una empresa. Por este motivo, es imprescindible tener perfectamente definido para cada empresa el concepto de satisfacción de sus clientes, desarrollando sistemas para su medición y creando modelos de respuesta inmediata ante la posible insatisfacción.”* (Pereiro, 2008).

“La satisfacción del cliente es la percepción que el cliente tiene sobre el grado en que se han cumplido sus requisitos. La satisfacción es un estado psicológico, y por tanto subjetivo, cuya obtención asegura fidelidad.” (Pereiro, 2008)

Entender con claridad el concepto de satisfacción del cliente tiene un gran significado, pero también resulta sumamente importante conocer los tipos de clientes que existen y no centrar toda la atención en el usuario final pues se correría el riesgo de fracasar.

Según la Norma ISO 9000:2005, *“el cliente es la organización o persona que recibe un producto, por lo tanto son los que tienen derecho de probarlo y luego evaluarlo”.* (Vivó, 2012)

Los clientes pueden ser clasificados como internos y externos.

Cliente Interno: *“es la red de servicios y productos internos que se proveen dentro de la organización, el cliente interno participa en la detección y corrección de defectos y errores.”* (Acevedo, y otros, 1995).

Cliente Externo: *“toda persona, grupo, empresa u organización que recibe algún resultado de la organización proveedora.”* (Acevedo, y otros, 1995).

En el ámbito de esta investigación, los clientes internos son los desarrolladores del software y los clientes externos son aquellos que se benefician de los sistemas creados.

Satisfacer las necesidades del cliente externo es un aspecto de vital importancia para las empresas, pero también se debe complacer al cliente interno porque la realización de su trabajo contribuye a que el producto desarrollado sea del agrado del usuario final.

Karl Albrecht en su libro Servicio al Cliente Interno dice: *“si desea que las cosas funcionen afuera, lo primero que debemos hacer es que funcionen adentro”.* Esto obliga a que las empresas desarrollen una cultura de servicio hacia lo interno de la empresa, si es que quieren ser reconocidas por sus clientes por la calidad del servicio que brindan. (Karl Albrecht, 1992)

Una empresa que desee lograr la realización de sus objetivos debe estar convencida del importante significado que tiene la satisfacción de sus clientes internos, pues su esfuerzo y dedicación determina la calidad del producto terminado.

Importancia de medir la satisfacción del cliente interno

La satisfacción de los clientes internos es determinante para el éxito o fracaso de cualquier empresa. Las organizaciones que se enfocan únicamente en satisfacer las necesidades de los clientes externos y no les prestan atención a los internos, están inevitablemente condenadas al fracaso. Conocer y superar las expectativas de estos, permite conseguir un mayor grado de responsabilidad laboral, de esta forma el Centro CEGEL podrá obtener beneficios tanto en términos humanos como económicos.

Se requiere un estudio de satisfacción del cliente interno cuando se necesite:

- Medir el nivel de satisfacción que existe.
- Anticipar conflictos asociados a problemas de insatisfacción que puedan derivar en la irremediable pérdida del personal que desarrolla el software.
- Conocer qué esperan los clientes internos que pueda mejorarse.
- Diagnosticar qué estrategias necesita implementar el Centro para aumentar la fidelidad del equipo de desarrollo.

A nivel mundial se han realizado varias investigaciones que se enfocan en el proceso de evaluación de la satisfacción del cliente interno, a continuación se hace referencia a algunos de estos trabajos, los cuales permitirán conocer la situación en que se encuentra este proceso en el mundo y en Cuba.

Evaluación de la satisfacción del cliente interno en el mundo y en Cuba

La evaluación de la satisfacción del cliente interno es el proceso mediante el cual se aplican técnicas para realizar la medición, se analizan los datos obtenidos para identificar las causas de insatisfacción y a partir de los aspectos evaluados insatisfactoriamente se toman medidas encaminadas a mejorarlos. En el mundo se ha avanzado en gran medida en cuanto a este tema, son muchas las organizaciones que miden la satisfacción de sus clientes externos, lo que no es tan habitual es utilizar esta misma práctica, pero trasladada al nivel interno de la organización, a esto se le llama satisfacción de cliente interno.

A continuación se muestran algunos ejemplos de cómo se mide la satisfacción del cliente interno en determinadas instituciones del mundo:

- Microsoft, IBM, Intel, Apple y SUN Microsystems miden la satisfacción de sus clientes internos mediante el uso de Glassdoor que es un sitio de Internet que permite a los empleados de diferentes empresas calificar, mediante encuestas anónimas, la empresa en la que trabaja cada uno. Su funcionamiento es simple: los empleados de estas empresas se registran anónimamente y hacen una pequeña encuesta donde se comprueba el grado de satisfacción que existe. La encuesta pide que los empleados califiquen su satisfacción con los factores laborales generales de la empresa, tales como las oportunidades de carrera, compensación y beneficios, el equilibrio entre trabajo y vida, la alta dirección, la cultura y los valores.
- Evaluación de la satisfacción del cliente interno en la Institución Prestadora de Servicios de la Salud (IPS) Oral Medic S.A: la empresa busca acrecentar su imagen frente a sus clientes externos y competidores. Adicionalmente tiene como objetivo el fortalecimiento de su clima laboral y por ende de su equipo de trabajo. La satisfacción del cliente interno es evaluada mediante la aplicación de una encuesta que pretende identificar su percepción respecto al clima laboral, identificando oportunidades de mejora. (Barón, 2010)
- En la organización Grupo Publicitario del Golfo se llevó a cabo una medición de la satisfacción del cliente interno porque se desconocían por completo las necesidades, expectativas y percepciones que se tienen sobre el ambiente laboral en el cual se desarrollan. Se realizó la aplicación de un cuestionario para aquellos miembros de la organización con una antigüedad mayor de dos años, y/o con contrato de trabajo por tiempo indefinido. El principal objetivo de este proceso de evaluación fue conocer los niveles de confort en el área de trabajo, el nivel de capacitación y el nivel de carga de trabajo. (Acuña, 2009)

En Cuba, en los últimos años, el tema de la satisfacción del cliente ha generado un gran debate entre las entidades que transitan por el proceso de Perfeccionamiento Empresarial. Lograr un nivel de ventas ha constituido la meta principal por alcanzar en las empresas y la satisfacción de los clientes ha quedado en un segundo plano. Sólo ahora se ha comenzado a orientar el rumbo hacia la satisfacción de los clientes, como punto de partida se comienzan a trazar estrategias que tengan en cuenta el nivel en que se encuentran satisfechas las necesidades de sus clientes externos. También se ha avanzado en cuanto a la

satisfacción de los clientes internos, aunque todavía resultan escasos los trabajos que se han realizado al respecto, pues la mayoría de las investigaciones efectuadas se han desarrollado en empresas dedicadas a la prestación de servicios mientras que en la industria del software no se ha otorgado la atención requerida a este tema. Seguidamente se muestran algunas evaluaciones de la satisfacción de los clientes internos realizadas en el país:

- Satisfacción por los servicios brindados en la sala de Fisioterapia del Hospital General “Dr. Juan Bruno Zayas Alfonso” de Santiago de Cuba: como instrumento de medición se utilizó la matriz de Fisher y como técnica la aplicación de encuestas, para lo cual se tuvieron en cuenta las variables: equipos, tecnología, procedimiento y relación entre los integrantes del personal, para ello se tomó una muestra de 63 trabajadores que laboraban en el departamento. (López, 2010)
- En el Centro de Información y Gestión Tecnológica de Matanzas se decidió evaluar la satisfacción del cliente interno. Se propone la evaluación a través de la aplicación del modelo SERVQUAL (Modelo de Calidad del servicio) que es un método estadístico con un alto nivel de fiabilidad y validez, que las empresas pueden utilizar para comprender mejor las expectativas y percepciones de los clientes respecto a un servicio. Para la evaluación del cliente interno se utiliza la técnica de las encuestas. Para ello se calculó la diferencia que existe entre las puntuaciones que asignan los empleados a las distintas parejas de declaraciones (percepciones-expectativas). (Cruz, Jerez, Álvarez, 2002)

En la UCI el tema de la satisfacción de los clientes internos se ha divulgado muy poco, se han realizado dos estudios relacionados con este proceso, el primero fue una evaluación realizada en la facultad 15 (actualmente facultad 3), donde se utilizó como técnica la aplicación de encuestas para la obtención del Índice de Satisfacción existente en los clientes internos vinculados a proyectos. La segunda investigación realizada logró definir los indicadores para medir la satisfacción de los clientes internos en el proceso de desarrollo de software de los proyectos del Centro de Gobierno Electrónico. La calidad de dicho proceso mejorará con la construcción de una herramienta informática que permita obtener la información de las encuestas en formato digital y que facilite la identificación de causas de insatisfacción como apoyo a la toma de decisiones administrativas en el Centro.

Para apoyar el proceso de evaluación de la satisfacción de los clientes internos, se han desarrollado en el mundo varios sistemas informáticos que facilitan la realización de dicho proceso, posteriormente se hace un estudio de algunos de ellos para determinar qué elementos pueden ser utilizados para la implementación de la herramienta que se desea obtener con esta investigación.

Sistemas informáticos que miden la satisfacción del cliente interno

En el mundo, existen varios sistemas informáticos que miden la satisfacción del cliente interno, a continuación se muestran algunos de ellos:

- Sistema de Medición Integrado de la Satisfacción Laboral: este sistema se desarrolla para la gestión de recursos humanos que labora en la industria de Maquila Textil ubicada en la Zona Franca de San Bartolo, El Salvador, con la finalidad de mejorar el clima organizacional existente. El instrumento que se emplea son las encuestas que miden los siguientes elementos: motivación, ambiente de trabajo y relaciones de trabajo. Con esta herramienta se logran identificar los puntos débiles de la empresa, obteniendo una visión clara y real de las condiciones laborales existentes en la organización.
- Glassdoor: es un sitio web creado en los Estados Unidos que permite a los empleados de las empresas compartir de forma anónima su opinión sobre el trabajo de su compañía. Utiliza la técnica de las encuestas para conocer sobre los salarios, las condiciones laborales y la opinión que tienen los clientes internos sobre las personas que los dirigen.
- Sistema integrado de gestión: procedimiento para medir la satisfacción de los funcionarios de la Universidad de Caldas frente a los servicios internos de los procesos. La medición de la satisfacción se realiza por medio de la aplicación de dos encuestas. Las encuestas se administran a través del Sistema Integrado de Gestión en el módulo de encuestas institucionales. De manera aleatoria el software le asigna una de las encuestas al personal seleccionado. Con la aplicación de estos instrumentos para medir la satisfacción se permitirá evaluar las condiciones institucionales para propósitos de re acreditación institucional. (Sistema Integrado de Gestión, 2009)

Los sistemas antes mencionados tienen como elemento común que utilizan como técnica la aplicación de encuestas y que los cuestionarios que se emplean incluyen preguntas relacionadas con las condiciones de trabajo y con la motivación. Estos sistemas informáticos no se pueden emplear para evaluar la

satisfacción de los clientes internos del Centro CEGEL porque están implantados en otros países y son software propietarios por lo que no se puede acceder a ellos, además no consideran los roles para la evaluación ni se tiene en cuenta la dimensión competencias técnicas que constituye un elemento imprescindible a medir en el Centro. Por otra parte, no se aplican en entornos de desarrollo de software por lo que no están dirigidos a los mismos objetivos y necesidades que se requieren, es por ello que se hace necesaria la creación de una aplicación que permita llevar a cabo este proceso. A continuación se realiza un estudio de las metodologías más utilizadas en la actualidad para seleccionar la que más se ajuste a las necesidades del sistema a implementar y utilizarla como guía en el ciclo de desarrollo.

1.3 Selección de la metodología de desarrollo

Lograr la construcción de un sistema informático eficiente, que cumpla con los requisitos planteados, es una tarea realmente intensa y sobre todo difícil de cumplir. Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y pesadas.

Programación Extrema (XP)

XP es una metodología de desarrollo de software enmarcada dentro del grupo de las ágiles y constituye la metodología más utilizada dentro de este grupo. Su objetivo principal es asegurar la producción de software con buena calidad y cubriendo las necesidades y requisitos del usuario. Esta metodología ofrece la posibilidad de cambiar los requisitos en cualquier momento de la vida de un proyecto, ya que es adaptable a estos cambios. Se centra en aumentar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo. XP busca la satisfacción del cliente tratando de mantener durante todo el tiempo su confianza en el producto. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final.

Proceso Unificado de Desarrollo (RUP)

RUP es una metodología para el desarrollo de software que define claramente quién, cómo, cuándo y qué

debe hacerse en el proyecto. Dicha metodología tiene tres características fundamentales: dirigido por casos de uso, iterativo e incremental y centrado en la arquitectura.

RUP generalmente es aplicado a grandes proyectos de desarrollo de software. Dentro de sus disciplinas gestiona el control de cambios, que permite mantener al equipo trabajando en los mismos artefactos, en cualquier momento del desarrollo del producto. RUP define como sus principales elementos a los trabajadores, las actividades, los artefactos y los flujos de actividades y divide el proceso de desarrollo de software en 4 fases fundamentales.

Selección de XP como metodología para guiar el ciclo de desarrollo de software

Se decidió utilizar la metodología de desarrollo XP porque vincula al cliente en el grupo de desarrollo y por las facilidades que brinda una metodología ágil a un proyecto de corta duración. Además, esta metodología brinda la posibilidad de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto, lo que constituye una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

1.4 Selección de herramientas, tecnologías, lenguajes, arquitectura y tipo de aplicación informática

Se decidió desarrollar una **aplicación web** porque se pueden ejecutar desde cualquier ordenador con conexión a internet, consumen muy pocos recursos del equipo en el que están instaladas y se pueden distribuir e instalar en varios equipos sin limitación o restricción alguna. El Centro CEGEL necesita que se pueda acceder a la herramienta desde cualquier lugar para que los clientes internos de los diferentes proyectos puedan responder las encuestas que son publicadas, por este motivo no se puede implementar una aplicación de escritorio porque el acceso a ellas solo se limita al ordenador donde están instaladas, además de que requieren instalación y actualización personalizada.

Herramientas de modelado

El modelado de sistemas de software es una técnica para tratar con la complejidad inherente a estos sistemas. El uso de modelos ayuda al ingeniero de software a visualizar el sistema a construir.

❖ **Rational Rose**

Es un software propietario utilizado para el modelado visual mediante UML.

Características de Rational Rose:

- Mantiene la consistencia de los modelos del sistema de software.
- Permite el chequeo de la sintaxis UML.
- Genera documentación automáticamente.
- Genera código a partir de los modelos.
- Permite realizar ingeniería inversa (crear modelo a partir código).
- Permite especificar, analizar y diseñar el sistema antes de codificarlo.(Blanco, 2012)

❖ **Visual Paradigm**

Es una potente herramienta CASE empleada para visualizar y diseñar elementos de software, para ello utiliza el lenguaje UML. Visual Paradigm proporciona a los desarrolladores una plataforma que les permite diseñar un producto con calidad de forma rápida. Provee el modelado de procesos de negocio, además de un generador de mapeo objetos-relacionales.

Características de Visual Paradigm:

- ✓ Se integra con diversos IDE´s de desarrollo como: NetBeans, Eclipse, Oracle JDeveloper, JBuilder.
- ✓ Soporta aplicaciones web.
- ✓ Genera código para Java y exportación como HTML.
- ✓ Es fácil de instalar y actualizar.
- ✓ Dispone de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
(Giraldo, 2005)

Después de haber realizado un estudio de las herramientas de desarrollo, en este caso Rational Rose y Visual Paradigm, se ha decidido que Visual Paradigm 8.0 es la herramienta que se empleará en la modelación de los procesos del negocio y en el diseño de la Base de Datos, se seleccionó por su característica de ser multiplataforma, también porque posee un lenguaje común que favorece la comunicación entre el equipo de desarrollo y además porque brinda apoyo adicional en cuanto a la generación de artefactos automáticamente.

Herramientas de desarrollo

Las herramientas de desarrollo ofrecen mejoras a la calidad y la productividad en el diseño y desarrollo.

❖ NetBeans 7.3

El IDE NetBeans es un entorno de desarrollo y es además, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Su código fuente está disponible para su reutilización. Se decidió utilizar este IDE porque es un producto de código abierto, gratuito y libre, para uso tanto comercial como no comercial y porque además, permite crear aplicaciones Web con PHP 5.

❖ Apache 2.2

El servidor web que se va a emplear para el desarrollo del sistema es Apache en su versión 2.2, se decidió utilizarlo por las siguientes razones: su rapidez facilita el análisis de los resultados de las encuestas, es gratuito y de código abierto por lo que se presenta como una alternativa de fácil acceso y además es altamente configurable, motivo por el cual puede ser adaptado a las necesidades del sistema que se va a implementar.

Tecnologías

❖ LimeSurvey 2.0

Para la implementación de la aplicación se va a utilizar LimeSurvey en la versión 2.0 que es una plataforma web gestionada por el Centro de Servicios de Informática y Redes de Comunicaciones de la Universidad de Granada, se decidió emplear esta tecnología porque fue diseñada especialmente para la administración de encuestas y en esta investigación se pretende utilizar como técnica para medir la satisfacción de los clientes internos la aplicación de encuestas. Además, es una aplicación de código abierto por lo que se puede emplear su código libremente y sin restricción alguna.

Una plataforma de servicios en línea es una página web que permite el acceso diferenciado de los usuarios según su nivel de responsabilidad para la realización de una tarea compartida: los usuarios con privilegios de administración tendrán acceso a las diferentes opciones de gestión de la herramienta (según

los permisos que se les haya conferido), sin embargo la mayoría de las personas que accederán a la página web únicamente podrán utilizar la herramienta (en este caso, para completar una encuesta), pero sin posibilidad de alterar ningún aspecto de la misma.

Para distinguir entre los diferentes roles se realiza una autenticación por medio de un nombre y una contraseña y será responsabilidad de la administración técnica de la plataforma el delegar los diferentes privilegios a los usuarios que sean creados.

LimeSurvey actualmente ofrece:

- Ilimitado número de encuestas simultáneas.
- Ilimitado número de preguntas en una encuesta.
- Ilimitado número de participantes en una encuesta.
- Encuestas multi-idioma.
- Encuestas anónimas y no anónimas.
- Amplia y amigable interfaz de administración.
- Soporte de más de 49 idiomas diferentes. (LimeSurvey)

Para desarrollar la aplicación web se pretenden aprovechar las características que ofrece LimeSurvey, pero además se hicieron modificaciones y se crearon nuevas funcionalidades, los cambios realizados se muestran a continuación:

- Cambios:
 - ✓ En la interfaz de la página principal del administrador.
 - ✓ En la interfaz de la página donde aparecen las encuestas disponibles.
 - ✓ En la interfaz de la página donde se filtran las estadísticas que se quieren conocer.
 - ✓ En la interfaz del panel central de participantes.
 - ✓ En las estadísticas en formato pdf y html.
- Eliminación de funcionalidades que no se requieren para el sistema que se va a desarrollar.
- Adición de las funcionalidades siguientes:
 - ✓ Calcular el Potencial de Satisfacción del Cliente Interno a nivel de rol.
 - ✓ Calcular y mostrar el Índice de Satisfacción del Cliente Interno por cada rol.

- ✓ Calcular y mostrar el Índice de Satisfacción del Cliente Interno de cada dimensión a nivel de rol.
- ✓ Calcular y mostrar el Índice de Satisfacción de cada elemento de una dimensión a nivel de rol.
- Adición de las gráficas donde se muestran las preguntas de cada encuesta con su Índice de Satisfacción.
- Se añadieron:
 - ✓ Las encuestas para cada rol.
 - ✓ Los participantes de cada encuesta.
- Se establecieron los permisos por encuesta.

Lenguajes

❖ PHP 5.3

Se va a emplear el lenguaje de programación PHP porque es de código abierto, interpretado, ampliamente usado y porque está diseñado especialmente para desarrollo web.

Ventajas de PHP:

- ✓ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destacando su conectividad con MySQL y PostgreSQL.
- ✓ Es un lenguaje multiplataforma.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite aplicar técnicas de programación orientada a objetos.

❖ HTML 5

Se va a hacer uso de HTML (Lenguaje de Marcas Hipertextuales) para describir y traducir la estructura y la información en forma de texto, así como para complementar dicho texto con imágenes. HTML permite la introducción de referencias a otras páginas por medio de los enlaces hipertexto y además con este lenguaje se puede modificar la apariencia de un documento hasta cierto punto, porque con CSS es que se complementa el diseño de la página.

❖ **CSS 3**

Las hojas de estilo en cascada constituyen un gran avance en el diseño web ya que permiten organizar el estilo y el diseño de las páginas. Se decidió emplear CSS porque este lenguaje incluye una amplia variedad de capacidades de estilo para satisfacer las necesidades de los diseñadores, además, permite más control sobre la apariencia del sitio. Las hojas de estilo en cascada ofrecen gran flexibilidad en términos de los efectos de la presentación tales como: margen, color, borde, fondo, entre otras.

❖ **Javascript 1.8.5**

Se determinó utilizar Javascript para la validación de forma básica. Además, se tomó esa decisión porque este lenguaje sirve principalmente para mejorar la gestión de la interfaz cliente/servidor y permite reconocer y manejar los eventos generados por el usuario. Estos eventos pueden ser el recorrido del propio documento HTML o la gestión de un formulario.

❖ **UML 2.1 (Lenguaje Unificado de Modelado)**

Como lenguaje de modelado se eligió UML en la versión 2.1 debido a que es un lenguaje gráfico que se utiliza para especificar o para describir métodos o procesos. Se usa para definir un sistema y para detallar los artefactos. UML no es un lenguaje de programación sino un lenguaje de propósito general para el modelado orientado a objetos y también puede considerarse como un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes, ya que ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura.

Arquitectura Modelo-Vista-Controlador (MVC)

La arquitectura MVC, está formada por tres niveles:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Se escogió la arquitectura MVC porque separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes. Además se tomó la decisión de utilizar esta arquitectura porque facilita la reutilización y permite una mejor organización de las clases.

1.5 Sistema Gestor de Base de Datos

Los Sistemas de Gestión de Base de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta.

Sistema Gestor de Base de Datos PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional ya que incluye características como son: la herencia, tipos de datos, funciones, restricciones y reglas. Su código fuente se encuentra disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

A continuación se muestran algunas de las características más importantes y soportadas por PostgreSQL:

- ✓ Integridad referencial.
- ✓ Juegos de caracteres internacionales.
- ✓ Regionalización por columna.
- ✓ Múltiples métodos de autenticación.
- ✓ Acceso encriptado vía SSL.
- ✓ Completa documentación.
- ✓ Disponible para Linux y UNIX en todas sus variantes y Windows 32/64bit.

Sistema Gestor de Base de Datos MySQL

MySQL es un sistema de gestión de bases de datos relacional, su diseño multihilo le permite soportar una gran carga de forma eficiente, es un gestor muy usado debido a su gran rapidez y facilidad de uso.

Características de MySQL:

- ✓ Consume muy pocos recursos, tanto de CPU como de memoria.
- ✓ Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.
- ✓ Funciona en diferentes plataformas.
- ✓ Posee un sistema de privilegios y contraseñas que es muy flexible y seguro, y permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está cifrado cuando se conecta con un servidor.
- ✓ Tiene soporte para comandos SQL para chequear, optimizar y reparar tablas.
- ✓ Soporta a grandes bases de datos.
- ✓ Probado con un amplio rango de compiladores diferentes.

Selección de MySQL como Sistema Gestor de Base de Datos

Luego de haber realizado un estudio de los Sistemas Gestores de Base de Datos (SGBD) PostgreSQL y MySQL, se decidió utilizar MySQL en su versión 5.5 debido a su gran rapidez, su facilidad de uso y porque se integra mucho mejor con LimeSurvey que es la tecnología que se va a emplear para el desarrollo del sistema. Además, porque facilita la gestión de usuarios y contraseñas y tiene soporte completo para operadores y funciones en las cláusulas de las consultas.

1.6 Técnicas, métricas y pruebas a emplear para la verificación y la validación del sistema

Para la verificación y la validación del sistema se hizo necesaria la aplicación de técnicas, métricas y pruebas, garantizando así que el software desarrollado cumpla los requisitos del cliente.

Técnicas de validación de requisitos

El proceso de validación de requisitos tiene por finalidad comprobar que los requisitos del software son consistentes, precisos, realistas y definen lo que el usuario desea del producto final.

La validación se realiza a través de diversos métodos. Los dos métodos más utilizados son:

Revisiones de requisitos: Consisten en una o varias reuniones planificadas, donde se intenta confirmar que los requisitos poseen los atributos de calidad deseados. Los requisitos son analizados sistemáticamente por un equipo de revisores. (Sommerville, 2005)

Construcción de prototipos: esta técnica consiste en la creación de una versión simplificada del sistema de software deseado, a partir de los requisitos recogidos en la especificación. Muestra, además de la interfaz, la secuencia de acciones que debe de realizar el sistema, aunque no en su totalidad. Los prototipos serán evaluados por el cliente y usuarios para comprobar su corrección y completitud. (Sommerville, 2005)

Métricas

“Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo del software y los proyectos de mantenimiento.” (Muñoz, 2006)

Para llevar a cabo la validación se utilizarán las siguientes métricas:

Métricas para Requisitos

En la disciplina Requisitos se tendrán en cuenta las métricas de estabilidad, especificidad y grado de validez de los requisitos; las cuales brindarán información sobre la comprensibilidad y validez de los mismos. (Fornaris, 2010)

Métricas para diseño

Para medir el diseño se utilizarán las métricas básicas inspiradas en el estudio de la calidad del diseño orientado a objeto que propone Pressman. Las métricas escogidas para la validación del diseño fueron la Métrica orientada a clases: Tamaño Operacional de Clase (TOC) y Acoplamiento entre Clases (AC). (Pressman, 1998)

Pruebas

“La prueba es el proceso de ejecutar un programa con el fin de encontrar errores”. (Departamento de Informática)

Pruebas unitarias

Las pruebas unitarias son las encargadas de verificar que el software cumple los requisitos de su especificación y además aseguran la calidad del código entregado. Son pruebas dirigidas a probar clases aisladamente y están relacionadas con el código y la responsabilidad de cada clase. (Rodríguez, 2006). Para aplicar las pruebas unitarias se van a emplear los métodos de pruebas de caja blanca y de caja negra.

➤ Método de prueba de caja blanca:

La prueba de caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener cada caso de prueba. Mediante el método de prueba de caja blanca, el ingeniero de software puede obtener casos de prueba que garanticen que se ejecuten por lo menos una vez todos los caminos independientes. (Pressman, 2008)

Para aplicar el método de prueba de caja negra se va a utilizar la **técnica de la partición equivalente** que divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El diseño de estos casos de prueba se basa en la evaluación de las clases de equivalencia.

➤ Método de prueba de caja negra:

Las pruebas de caja negra se centran en los requisitos funcionales del software, o sea, la prueba de caja negra permite al ingeniero de software obtener conjuntos de condiciones de entrada que ejecuten completamente todos los requisitos funcionales de un programa. La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores. La prueba de caja negra intenta encontrar errores de las siguientes categorías: funciones incorrectas, errores de interfaz, etc. (Pressman, 2008)

Para emplear el método de prueba de caja blanca se decidió emplear la **técnica del camino básico** que permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez.

Pruebas de aceptación

Las pruebas de aceptación se van a poner en práctica a través de los casos de prueba de aceptación que propone la metodología de desarrollo XP y con ellos se podrá confirmar que cada Historia de Usuario fue

implementada correctamente. Estas pruebas son destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente final.

1.7 Conclusiones del capítulo

En este capítulo se concluye lo siguiente:

- El estudio realizado permitió definir los conceptos asociados a calidad de software y satisfacción del cliente, vinculando ambos términos y determinando que la investigación se va a enfocar específicamente en el proceso de evaluación de la satisfacción del cliente interno.
- La medición de la satisfacción del cliente interno tiene tendencias en el mundo, en Cuba y en la UCI, la eficiencia de este proceso puede estar apoyado por la informática.
- Con el análisis que se realizó de la satisfacción del cliente interno a nivel mundial se constató que en Cuba y específicamente en la UCI este proceso no se ha explotado lo suficiente, demostrando de esta manera la necesidad de la creación de sistemas informáticos que faciliten la realización de dicho proceso.
- Se propone para evaluar la satisfacción del cliente interno en el Centro de Gobierno Electrónico de la Facultad 3 la realización de una aplicación web que permita llevar a cabo este proceso.
- Se escogieron para este contexto herramientas, lenguajes y tecnologías atendiendo a diseño, entorno de desarrollo y gestión de datos que cumplan las necesidades de la aplicación a desarrollar. Además, se seleccionó la metodología a utilizar como guía en el ciclo de desarrollo de software.
- Se plantea la utilización de métricas, técnicas de validación y pruebas para comprobar el funcionamiento de la herramienta que se construya.

Capítulo 2: Análisis, diseño e implementación de la propuesta de solución

CAPÍTULO 2: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN

Introducción

En este capítulo se identifican los requisitos funcionales y no funcionales que la aplicación a desarrollar debe cumplir, además, se generan los artefactos de las fases de Planificación, Diseño y Codificación, propias de la metodología de desarrollo XP, donde se confeccionan las Historias de Usuario para describir el comportamiento del sistema y se planifica estimando el tiempo de desarrollo de cada una. Además, se identifica el patrón arquitectónico y los patrones de diseño que se van a utilizar. Por último se procede a codificar definiendo estándares de codificación para un mejor entendimiento.

2.1 Definición de dimensiones, roles, atributos y método de medición para evaluar la satisfacción de los clientes internos

En la tesis de diploma presentada en el año 2011 titulada *“Definición de indicadores para medir la satisfacción de los clientes internos en el proceso de desarrollo de software de los proyectos del Centro de Gobierno Electrónico de la Facultad 3”* se precisaron las dimensiones, los atributos, los roles vinculados al proceso de formación y el método de medición a utilizar para evaluar la satisfacción de los clientes internos.

Las dimensiones son condiciones que se deben establecer con el objetivo de conocer dónde se encuentran los principales problemas de una institución, que no le permiten alcanzar una satisfacción laboral máxima. Estas condiciones son percibidas por los empleados en función de sus necesidades individuales y de las características del trabajo que desempeñan. En función de estas dimensiones se pueden elaborar estrategias de mejora para dar solución a los inconvenientes encontrados. Estas, constituyen un punto de partida importante en la investigación cuyo papel principal es, agrupar de forma lógica y coherente, los parámetros a evaluar para la obtención del nivel de satisfacción laboral existente en los proyectos del Centro. (Noa, 2011)

Para conocer el Índice de Satisfacción del Cliente Interno (ISCI) se van a utilizar las siguientes dimensiones:

- Condiciones de Trabajo: son las características que se refieren tanto a la apariencia del local como a los medios básicos de trabajo.

- Organizativas: se refiere a aspectos relacionados con la organización existente dentro de la entidad.
- Motivación: son todas aquellas acciones capaces de estimular al personal de la entidad a la realización de actividades.
- Empatía: se refiere a la atención personalizada que ofrecen las instituciones a sus trabajadores.
- Competencias Técnicas: abarca las características esencialmente técnicas de los informáticos. (Noa, 2011)

Roles

Se determinan 5 roles vinculados al proceso de formación:

1. Rol Analista.
2. Rol Desarrollador (programador).
3. Rol Diseñador de bases de datos.
4. Rol Revisor técnico.
5. Rol Administrador de la configuración. (Noa, 2011)

Atributos

Los atributos son características que se desean evaluar con el fin de conocer si su funcionamiento y desenvolvimiento dentro de una entidad es el más eficiente posible. Son características que de una forma u otra afectan tanto al personal de una empresa como a los productos que se elaboran dentro de ella.

Los atributos a emplear van a estar enmarcados dentro de las 5 dimensiones definidas, según el tipo de información que van a tratar. Para el caso de la dimensión identificada con el nombre Competencias Técnicas los atributos que la integran serán estratificados por roles. Estos, son redactados en función de las competencias técnicas establecidas para cada rol y teniendo en cuenta algunos aspectos que pueden provocar disgustos en los clientes internos desde el rol que ejercen. Los atributos que forman parte de las restantes 4 dimensiones se elaboran a partir de características cuyo comportamiento puedan ser posibles causas de insatisfacción. Se identifican un total de 61 atributos. (Noa, 2011)

Atributos que integran la dimensión Condiciones de Trabajo:

- Ventilación y ambientación del local.
- Iluminación del local.
- Áreas de trabajo seguras.
- Distribución de equipos, muebles y espacios.

- Limpieza de los equipos y del local de trabajo.
- Estado técnico de los medios de trabajo.
- Ambiente laboral.
- Características de las máquinas. (Noa, 2011)

Atributos que integran la dimensión Condiciones Organizativas:

- Adecuada asignación de roles.
- Relación asignación de tareas – tiempo disponible.
- Realización de tareas.
- Horarios de trabajo. (Noa, 2011)

Atributos que integran la dimensión Motivación:

- Reconocimiento del trabajo realizado.
- Capacitación y superación.
- Solución de problemas en conjunto.
- Relación docencia – producción.
- Actividades extra-producción. (Noa, 2011)

Atributos que integran la dimensión Empatía:

- Directivos comprensivos.
- Preocupación y respuesta a las inquietudes. (Noa, 2011)

Atributos que integran la dimensión Competencias técnicas para el Rol Analista:

- Obtención de requisitos.
- Descripción de requisitos.
- Gestión y trazabilidad de requisitos.
- Cambios en requisitos validados.
- Gestión del proyecto con especialistas capacitados.
- Uso de herramientas actualizadas.
- Esclarecimiento de dudas.
- Cambio de proveedores de requisitos. (Noa, 2011)

Atributos que integran la dimensión Competencias técnicas para el Rol Desarrollador:

- Aceptación de los lenguajes de programación.

- Disponibilidad de los servidores.
- Últimas versiones de código.
- Uso de herramientas.
- Uso de la lógica y algoritmos de programación.
- Uso de patrones.
- Cambios en la implementación.
- Artefactos necesarios. (Noa, 2011)

Atributos que integran la dimensión Competencias técnicas para el Rol Diseñador de BD:

- Modelo lógico de la base de datos.
- Modelo físico de la base de datos.
- Normalización.
- Trabajo con la base de datos.
- Disponibilidad de la base de datos.
- Uso de herramientas.
- Lenguajes de implementación.
- Transformación de diagrama.
- Configuración de roles y usuarios. (Noa, 2011)

Atributos que integran la dimensión Competencias técnicas para el Rol Revisor técnico:

- Elaboración de artefactos.
- Disponibilidad de los casos de pruebas.
- Capacitaciones según la aplicación a revisar.
- Pruebas de caja blanca.
- Disponibilidad de los servidores.
- No conformidades.
- Disponibilidad de la aplicación.
- Comportamiento de la aplicación.
- Corrección del caso de prueba. (Noa, 2011)

Atributos que integran la dimensión Competencias técnicas para el Administrador de la configuración:

- Uso de herramientas.
- Plan de gestión de la configuración.
- Auditorías.
- Políticas de gestión de la configuración.
- Configuración de las herramientas.
- Creación y liberación de líneas bases.
- Esclarecimiento de dudas. (Noa, 2011)

Método de medición

El método de medición que se va a utilizar para evaluar la satisfacción de los clientes internos son las encuestas (Noa, 2011)

Se tomó la decisión de utilizar las encuestas como método de medición por las siguientes razones: tienen un coste muy bajo, son la forma más discreta de recogida de datos y pueden ofrecer a los encuestados completo anonimato, especialmente para aquellos clientes que no desean realizar entrevistas personales o telefónicas. Como la respuesta es anónima, no hay problemas con la naturaleza sensible o embarazosa de las preguntas del mismo.

La encuesta estará conformada por 5 cuestionarios, uno para cada rol, los cuales van a tener un total de 25 elementos a evaluar, cada elemento se evalúa a partir de una escala de valores del 1 al 5, donde cada número significa:

1. Me siento insatisfecho con este aspecto.
2. Me molesta un poco este aspecto.
3. No me siento ni satisfecho ni insatisfecho con este aspecto.
4. Me siento un poco satisfecho con este aspecto.
5. Me siento totalmente satisfecho con este aspecto. (Noa, 2011)

Con los elementos antes mencionados se pretende evaluar la satisfacción de los clientes internos en el proceso de desarrollo de software de los proyectos del Centro CEGEL, para ello se va a desarrollar una herramienta que permita llevar a cabo dicho proceso de evaluación.

2.2 Descripción del sistema

Escenario actual del proceso de evaluación de la satisfacción del cliente interno en el Centro

El Asesor de Calidad decide evaluar la satisfacción de los clientes internos en el proceso de desarrollo de software de los proyectos del Centro, para ello confecciona un conjunto de encuestas y las reparte en los diferentes proyectos para que sean contestadas por cada uno de sus integrantes, luego recibe las respuestas y procede a realizar de manera manual el análisis de los datos, por último entrega los resultados obtenidos a la dirección del Centro para que se tomen medidas encaminadas a mejorar los aspectos donde se encuentran insatisfechos los clientes internos.

Para ver el diagrama del proceso, consultar [anexo](#).

Propuesta del sistema

La solución propuesta está diseñada para los proyectos del Centro de Gobierno Electrónico de la Facultad # 3. El objetivo de esta herramienta es evaluar la satisfacción de los clientes internos en el proceso de desarrollo de software y de esta forma identificar las causas que le producen insatisfacción a los clientes internos del Centro, para ello se hace necesaria la implementación de funcionalidades para la gestión de usuarios, para la gestión de encuestas y para el reporte de los resultados de las encuestas.

Con la herramienta que se pretende desarrollar se podrá evaluar la satisfacción de los 5 roles definidos, para ello se aplicará una encuesta en correspondencia con el rol. Si posteriormente se desean agregar nuevos roles al proceso de evaluación, se deben determinar las fórmulas para calcular el Índice de Satisfacción y además, se deben confeccionar las encuestas para esos roles porque las que se elaboraron en el trabajo de diploma presentado en el año 2011 solo incluyen al analista, al desarrollador, al diseñar de BD, al revisor técnico y al administrador de la configuración, es válido aclarar que esta investigación solo se va a centrar en la evaluación de la satisfacción de esos roles.

Requisitos del cliente

Un requisito es una característica que el sistema debe tener para ser aceptado por el cliente. A continuación se definen los requisitos funcionales y no funcionales que la aplicación a desarrollar debe cumplir.

Requisitos funcionales

Los requisitos funcionales (RF) definen los servicios que el sistema debe proporcionar, cómo debe reaccionar a una entrada particular y cómo se debe comportar ante situaciones particulares, es decir, describen el funcionamiento del sistema.

El sistema debe permitir:

RF1: Autenticar usuario

RF2: Adicionar usuario

RF3: Listar usuarios

RF4: Modificar usuario

RF5: Establecer permisos de usuario

RF6: Eliminar usuario

RF7: Adicionar encuesta

RF8: Adicionar grupo de preguntas

RF9: Modificar grupo de preguntas

RF10: Adicionar pregunta

RF11: Modificar pregunta

RF12: Eliminar pregunta

RF13: Listar encuestas

RF14: Eliminar encuesta

RF15: Mostrar Índice de Satisfacción del Cliente Interno a nivel de rol

RF16: Mostrar Índice de Satisfacción de cada dimensión a nivel de rol

RF17: Mostrar Índice de Satisfacción de cada elemento de una dimensión a nivel de rol

Requisitos no funcionales

Los requisitos no funcionales (RNF) son restricciones de las funciones ofrecidas por el sistema. Generalmente se aplican al sistema en su totalidad y surgen de las necesidades del usuario. Pueden especificar también la utilización de una herramienta en particular o un lenguaje de programación.

Usabilidad

- Tipo de usuario Final.

RNF1: el sistema será usado por el Asesor de Calidad, por la Dirección del Centro CEGEL y por los clientes internos que pertenezcan a dicho Centro.

- Tipo de aplicación informática:

RNF2: el producto será una aplicación web enfocada en las necesidades del Centro de Gobierno Electrónico para evaluar la satisfacción de los clientes internos.

- Finalidad:

RNF3: este sistema estará enfocado en el proceso de evaluación de la satisfacción de los clientes internos en el proceso de desarrollo de software de los proyectos del Centro de Gobierno Electrónico con la finalidad de identificar las causas que les producen insatisfacción a los clientes internos.

- Ambiente:

RNF4: el sistema deberá tener una interfaz de usuario agradable y fácil de usar.

RNF5: agrupar vínculos y botones por grupos funcionales. La consistencia de la interacción entre usuario y sistema estará determinada por el diseño de la interfaz de usuario que mantendrá los elementos como menús, banners y zona de trabajo en posiciones fijas, además presentará la mayor uniformidad posible entre cuadros de texto y botones.

RNF6: mostrar los mensajes, títulos y demás textos que aparezcan en la interfaz del sistema en idioma español. Tanto los títulos de los componentes de la interfaz, como los mensajes para interactuar con los usuarios, así como los mensajes de error, deberán ser en idioma español y tener una apariencia uniforme en todo el sistema. Los mensajes de error deberán ser lo suficientemente informativos para dar a conocer la severidad del error.

Eficiencia

RNF7: el sistema debe ser capaz de dar respuestas a las peticiones con un nivel aceptable de desempeño. Teniendo en cuenta el nivel de concurrencia que pueda existir, debe ser capaz de prestar servicio sin que se deterioren los tiempos de respuestas.

RNF8: el sistema debe garantizar el acceso concurrente para todos los usuarios del sistema en el tiempo que se habilite para la evaluación.

Soporte

RNF9: se deberá recibir entrenamiento para la configuración y el mantenimiento del sistema.

Restricciones de diseño

RNF10: para el montaje del sistema se requerirá el servidor web Apache 2.2, el sistema gestor de base de datos MySQL 5.5 y la plataforma web LimeSurvey para la administración de encuestas.

Apariencia o Interfaz Externa

RNF11: el sistema presentará una interfaz amigable y fácil de usar.

Seguridad

RNF12: el sistema podrá ser utilizado solamente por usuarios autenticados en el mismo.

RNF13: el sistema brindará la posibilidad de establecer permisos sobre acciones, garantizando que solo acceda a la información quien esté autorizado.

Requisitos de software

RNF14: instalar en las estaciones de trabajo que van a acceder a la aplicación los elementos necesarios para el correcto funcionamiento del sistema:

- Microsoft Internet Explorer 6.0 o superior o Mozilla Firefox 2.0 o superior.

RNF15: instalar en la estación de trabajo donde va a estar ubicada la aplicación lo siguiente:

- Sistema gestor de base de datos MySQL 5.5.

- Servidor web Apache 2.2.

Requisitos de hardware

RNF16: proporcionar características mínimas de hardware a las estaciones de trabajo.

Las características técnicas mínimas de hardware deben ser las siguientes:

- 1 GB de RAM
- 20 GB de disco duro.
- Microprocesador Intel Core 2 Duo.

Personas relacionadas con el sistema

Se define como persona relacionada con el sistema aquel que administra todos los procesos que se llevan a cabo en la aplicación y también forman parte de este grupo aquellos usuarios que utilizan el sistema para responder una determinada encuesta.

El **Administrador** (en este caso, el Asesor de la Calidad) es la persona encargada de gestionar todos los procesos que se llevan a cabo en el sistema para evaluar la satisfacción de los clientes internos y obtiene los resultados de las encuestas realizadas a partir de las dimensiones y atributos definidos.

Los usuarios que están autorizados a utilizar el sistema solo para responder encuestas son los siguientes:

- Analista.
- Desarrollador.
- Diseñador de Base de Datos.
- Revisor técnico.
- Administrador de la configuración.

Historias de usuario

Las historias de usuario (HU) representan una breve descripción del comportamiento del sistema, emplea terminología del cliente sin lenguaje técnico, se realiza una por cada característica principal del sistema y se utilizan para hacer estimaciones de tiempo, además reemplazan un gran documento de requisitos.

A continuación se muestra una HU de prioridad alta en el negocio, el resto se puede encontrar en el expediente de proyecto y en los [anexos](#):

HU #2: Adicionar usuario

Historia de Usuario	
Número: 2	Usuario: Administrador
Nombre de historia: Adicionar usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1 semana	Iteración asignada: 1
Programador responsable: Yaima Morales Labrada	
Descripción: El administrador accede a la interfaz de la aplicación que le permite la adición de usuarios, llena los siguientes campos: nombre de usuario, correo electrónico y nombre completo, por último selecciona la opción "Añadir usuario".	
Observaciones: Si no se llenan todos los campos o se introduce algún dato incorrecto, el sistema muestra un mensaje de error señalando que no se pudo adicionar el usuario porque se dejó un campo vacío o el campo introducido no es válido.	

Tabla #1: Historia de Usuario

2.3 Planificación

En esta fase el cliente establece la prioridad de cada Historia de Usuario y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Las estimaciones de esfuerzo asociadas a la implementación de las historias las establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos.

Estimación de esfuerzo por Historias de Usuario

Historia de Usuario	Puntos de estimación (semanas)
1 Autenticar usuario	1
2 Adicionar usuario	1
3 Listar usuarios	1
4 Modificar usuario	1
5 Establecer permisos de usuario	2
6 Eliminar usuario	1
7 Adicionar encuesta	2
8 Adicionar grupo de preguntas	1
9 Modificar grupo de preguntas	1
10 Adicionar pregunta	1
11 Modificar pregunta	1
12 Eliminar pregunta	1
13 Listar encuestas	2
14 Eliminar encuesta	1
15 Mostrar Índice de Satisfacción del Cliente Interno a nivel de rol	3
16 Mostrar Índice de Satisfacción de cada dimensión a nivel de rol	3
17 Mostrar Índice de Satisfacción de cada	3

elemento de una dimensión a nivel de rol	
--	--

Tabla #2: Estimación de esfuerzo por Historias de Usuario

Plan de iteraciones

Luego de haber identificado las HU y de realizar una estimación del tiempo requerido para la realización de cada una, se lleva a cabo la planificación estableciendo las iteraciones que se necesitarán antes de entregar el sistema.

Se decidió dividir la implementación de las HU en dos iteraciones:

Iteración 1:

En la iteración 1 se llevará a cabo el desarrollo de las HU del número 1 hasta el número 14.

Iteración 2:

En la iteración 2 se llevará a cabo el desarrollo de las HU del número 15 hasta el número 17.

Plan de duración de las iteraciones

El plan de duración de las iteraciones se encarga de mostrar las HU en el orden en que se implementarán en cada iteración, así como el tiempo de duración que se estima va a demorar su desarrollo.

Iteración	Orden de la Historias de usuario a implementar.	Duración total
1	Autenticar usuario Adicionar usuario Listar usuarios Modificar usuario Establecer permisos de usuario	6 semanas

	Eliminar usuario Adicionar encuesta Adicionar grupo de preguntas Modificar grupo de preguntas Adicionar pregunta Modificar pregunta Eliminar pregunta Listar encuestas Eliminar encuesta	
2	Mostrar Índice de Satisfacción del Cliente Interno a nivel de rol Mostrar Índice de Satisfacción de cada dimensión a nivel de rol Mostrar Índice de Satisfacción de cada elemento de una dimensión a nivel de rol	4 semanas

Tabla #3: Plan de duración de las iteraciones

Plan de entregas

El plan de entregas especifica la fecha fin de cada iteración y las Historias de Usuario que deberán estar implementadas para esa fecha.

Fin de la Iteración 1	Fin de la Iteración 2
------------------------------	------------------------------

(30 de marzo del 2013)	(30 de abril del 2013)
Historias de usuario (1-14)	
	Historias de usuario (15-17)

Tabla #4: Plan de entregas

2.4 Diseño

La metodología X.P sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño entendible y de fácil implementación que a la larga costará menos tiempo y esfuerzo desarrollar. Como parte de esta fase, se define el patrón arquitectónico a utilizar para solucionar problemas de arquitectura y además se precisan los patrones de diseño que se van a emplear para remediar problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño.

Patrón de Arquitectura

Un patrón de arquitectura de software es un esquema genérico probado para solucionar un problema particular, el cual es recurrente dentro de un cierto contexto. Este esquema se especifica describiendo los componentes, con sus responsabilidades y relaciones. (Ochoa, 2005). Para el desarrollo del presente sistema el patrón de arquitectura que se utilizará es el Modelo-Vista-Controlador (MVC).

Patrón MVC

El patrón MVC es un patrón de arquitectura que separa la lógica de negocio de la interfaz de usuario, facilita la evolución por separado de ambos aspectos e incrementa la reutilización y flexibilidad. Divide una aplicación interactiva en tres partes: el modelo contiene los datos y la funcionalidad esencial, las vistas despliegan la información al usuario y los controladores manejan las entradas. Es una especialización de un modelo de capas, con la diferencia que se usa para entornos web como patrón por excelencia.

Modelo:

Es la capa encargada de encapsular toda la lógica de negocio de la aplicación. Esta capa se puede subdividir en varias:

- Lógica de negocio: contiene clases para constituir lo referente a la capa de aplicación, se encarga de atender las peticiones de los controladores y así dar una respuesta acorde con lo recibido.
- Capa de datos: se encarga de gestionar toda la interconexión con el Sistema Gestor de Base de Datos, solo se comunica con la lógica de negocio.
- Helpers: llamados “ayudantes” apoyan tanto al controlador como a la vista para hacer más livianas algunas tareas.

Vista:

Es la respuesta de cada controlador y lo que se le presenta al usuario final, se puede comunicar con el controlador, los “helpers” y el modelo (en algunas ocasiones).

Controlador:

Es el eje central de la arquitectura, encargada de gestionar todas las peticiones, validar las entradas recibidas y dirigir cualquier petición de cualquier tipo. Solo se comunica con el modelo y responde a través de vistas. (Gómez, 2011)

Clases a utilizar

A continuación se definen algunas de las clases que se van a utilizar en el diseño del sistema, se presenta una breve explicación de la responsabilidad que tiene cada una y se dividen teniendo en cuenta las capas de la arquitectura MVC.

Modelo:

Usuario: clase que gestiona la conexión a la base de datos donde se guardan los usuarios (en este caso roles) y contraseñas registrados, además de cada uno se almacena un identificador determinado.

Encuesta: clase que gestiona la conexión a la base de datos donde se guarda el identificador (id) y el nombre de las encuestas creadas.

Grupo: clase que gestiona la conexión a la base de datos donde se guarda el id del grupo, el id de la encuesta a la que corresponde dicho grupo y el nombre del mismo.

Pregunta: clase que gestiona la conexión a la base de datos donde se guarda el id de la pregunta, el id de la encuesta a la que corresponde dicha pregunta, el id del grupo al que pertenece y el texto de las preguntas.

Respuesta: clase que gestiona la conexión a la base de datos donde se guarda el id de la respuesta, la puntuación dada y el id de la pregunta a la que pertenece dicha respuesta.

Participante: clase que gestiona la conexión a la base de datos donde se guarda el id que le corresponde a un participante determinado, además se almacena el id de la encuesta que respondió, el id de la respuesta dada y el id del usuario que le pertenece.

Vista:

Autenticar: clase que muestra la interfaz de usuario para la autenticación del sistema.

ModificarUsuario: clase que muestra la interfaz de usuario que permite modificar un usuario determinado.

ListaEncuestas: clase que muestra la interfaz de usuario donde aparece la lista detallada de las encuestas que han sido creadas.

Estadísticas_View: clase que muestra la interfaz de usuario donde se exponen las estadísticas de las encuestas.

ModificarPregunta: clase que muestra la interfaz de usuario que permite modificar una pregunta determinada.

AdicionarGrupo: clase que muestra la interfaz de usuario que permite adicionar un grupo a una encuesta.

NuevaEncuesta: clase que muestra la interfaz de usuario que permite adicionar una nueva encuesta.

Controlador:



Autenticación: clase que se encarga de procesar la autenticación de los usuarios en el sistema.

AccionesUsuario: clase que se encarga de procesar todas las funcionalidades asociadas a la gestión de usuarios.

AdministraciónEncuestas: clase que se encarga de procesar todas las funcionalidades asociadas a la gestión de encuestas.

Estadísticas: clase que se encarga de procesar las estadísticas de las encuestas realizadas.

Patrones de Diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. (Gamma, 1995). El uso de patrones ayuda a obtener un software de calidad. Los patrones de diseño representan soluciones a problemas que surgen al desarrollar software en un cierto contexto, capturan la estructura estática y dinámica y la colaboración entre los participantes claves en diseños de software y facilitan la reutilización.

Patrones GRASP

La asignación de responsabilidades es la habilidad más importante en el análisis y diseño orientado a objetos. Respetar los principios fundamentales es uno de los factores críticos para obtener diseños reutilizables, mantenibles y extendibles.

Experto: el patrón Experto indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo.

Las clases Usuario, Encuesta, Grupo, Pregunta, Respuesta y Participante cuentan con la información necesaria para cumplir cada una de las responsabilidades que les corresponden, por tanto, el patrón experto se evidencia en cada una de estas clases.

Creador: el patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos.

Las clases Autenticación, AccionesUsuario, AdministraciónEncuestas y Estadísticas son responsables de la creación de nuevos objetos y clases.

Controlador: el patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según corresponda.

Las clases Autenticación, AccionesUsuario, AdministraciónEncuestas y Estadísticas serán las responsables de atender los eventos del sistema y además, son las encargadas de capturar las llamadas que se efectúan en la aplicación en tiempo real.

Bajo Acoplamiento: es la idea de tener las clases lo menos ligadas entre sí que se pueda, de tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre ellas.

En todo el diseño del sistema se emplea el patrón Bajo Acoplamiento ya que solo se encuentran relacionadas entre sí las clases necesarias y se trata siempre de que estén ligadas la menor cantidad posible para que en caso de ocurrir un cambio en una de ellas, las otras no se vean implicadas.

Alta cohesión: la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase.

En todo el diseño del sistema se emplea el patrón Alta Cohesión ya que las funcionalidades de una clase comparten su responsabilidad de una operación con otras funcionalidades, siempre que sea necesario, contribuyendo así a la alta reutilización y a una menor complejidad de implementación.

Patrones GOF (Pandilla de los cuatro)

Clasificación de los patrones GOF según su propósito:

De creación: forma de crear instancias.

(Factoría (Factory), Singular (Singleton)).

Estructurales: cómo se combinan clases y objetos para formar nuevas estructuras y proporcionar nuevas funciones (Adaptador (Adapter), Decorador (Decorator)).

De comportamiento: ayudan a definir la comunicación e interacción de los objetos.

(Observador (Observer)).

En el diseño del sistema sólo se utiliza el patrón GOF estructural Decorador.

Decorador

El patrón Decorador responde a la necesidad de añadir dinámicamente funcionalidad a un objeto. Esto permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera.

Este patrón se pone de manifiesto en todas las clases que muestran la vista, un ejemplo de ello se evidencia en la interfaz principal del administrador que tiene un marco común y de él heredan otras interfaces que se muestran en dependencia de la acción que este decida realizar.

Tarjetas CRC (Clase – Responsabilidad – Colaborador)

La metodología de desarrollo XP como parte de la fase de diseño propone el modelado CRC. Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores.

A continuación se muestra la tarjeta CRC de una de las clases controladoras con las que cuenta el sistema, las tarjetas restantes se encuentran disponibles en el expediente de proyecto y en los [anexos](#).

Clase AdministracionEncuestas

Clase AdministracionEncuestas	
Responsabilidad	Colaborador
Adicionar encuesta	Encuesta
Adicionar grupo de preguntas	Grupo
Modificar grupo de preguntas	Grupo
Adicionar pregunta	Pregunta
Modificar pregunta	Pregunta
Eliminar pregunta	Pregunta

Listar encuestas	Encuesta
Eliminar encuesta	Encuesta

Tabla # 5: Tarjeta CRC Clase AdministracionEncuestas

Diseño de la base de datos

El modelo de datos se encuentra compuesto por un conjunto de tablas nombradas con el prefijo tb_.

- En la tabla tb_usuario se registrarán todos los usuarios y contraseñas que son creados por el administrador, además se almacenará un id para cada usuario.
- La tabla tb_encuesta tendrá el id y el nombre de cada encuesta creada.
- La tabla tb_grupo recopila el id del grupo, el nombre y el id de la encuesta a la que pertenece dicho grupo.
- La tabla tb_pregunta por su parte almacena el id de la pregunta, el texto, el id de la encuesta y el id del grupo al que corresponde dicha pregunta.
- En tb_respuesta se guarda el id de la respuesta, la puntuación dada y el id de la pregunta a la que pertenece dicha respuesta.
- En la tabla participante se registra el id que le corresponde a un participante determinado, además se almacena el id de la encuesta que respondió, el id del usuario que le pertenece y el id de la respuesta dada por dicho participante.

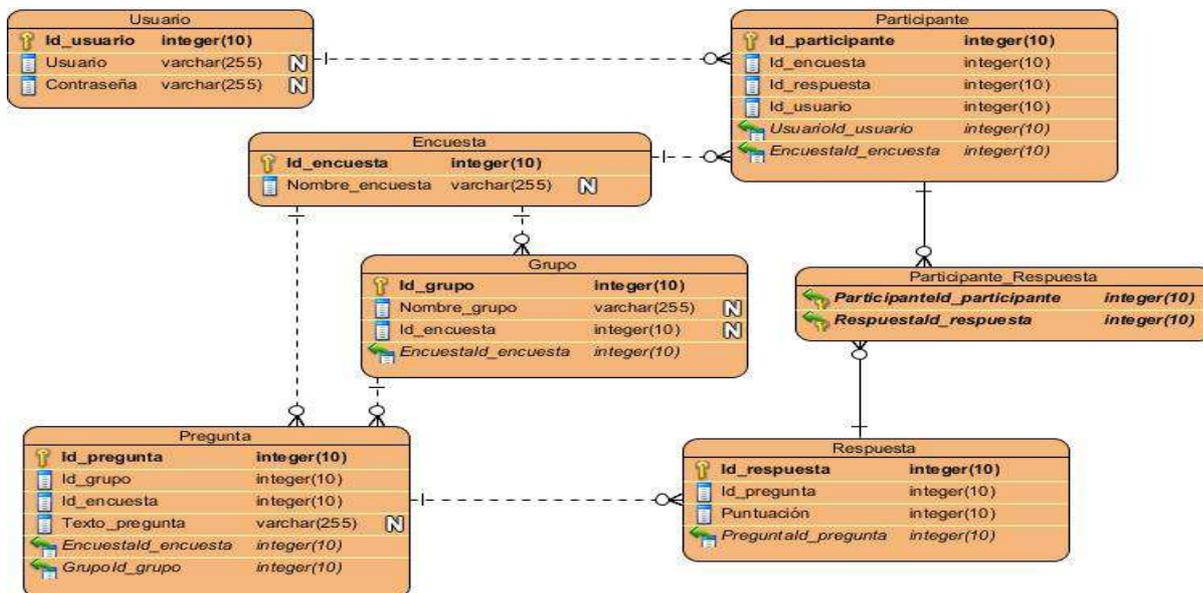


Figura #1: Modelo Físico de la Base de Datos.

2.5 Codificación

Es necesario codificar y plasmar las ideas a través del código. En programación, el código expresa la interpretación del problema, se utiliza para comunicar, para hacer comunes las ideas y por tanto para aprender y mejorar.

La metodología XP propone que en la fase de codificación, el cliente y el desarrollador del proyecto deben estar en comunicación para que el programador pueda codificar todo lo necesario para el proyecto que se requiere, en esta parte está incluido todo lo relacionado con la programación.

La codificación debe hacerse teniendo en cuenta estándares de codificación ya definidos. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

Estándares de codificación

Se definen estándares de codificación porque un estilo de programación homogéneo ofrece la posibilidad de que cualquier persona que consulte el código lo pueda entender en menos tiempo y que el código en consecuencia sea mantenible.

Con vistas a garantizar la homogeneidad de dicho código, se establece el estilo descrito a continuación:

Comentarios:

- Se utilizan comentarios para describir el código. Sintaxis: //

Declaraciones:

- Número de declaraciones por línea:

Se declara cada variable en una línea distinta, de esta forma cada variable se puede comentar por separado. Ejemplo:

```
$am =0; // media aritmética
```

- Variables:

Las variables se declaran de la siguiente forma:

Ej: \$cont =0;

Los arrays se declaran de este modo:

```
$arrayDimensiones = array();
```

- Declaraciones de clases y funcionalidades:

Se siguen estas reglas:

No hay espacio entre el nombre del método, el paréntesis y la lista de parámetros.

Se abre la llave al final de la misma línea de la declaración de las clases y de los métodos.

Sentencias:

- Sentencias simples.

Cada línea contiene una única sentencia. Ejemplo:

```
$dimension = array_sum($this->arrayPromedios) / 5;
```

- Sentencias compuestas:

Son las que están entre llaves. Todas las sentencias del tipo if, for, while o switch tienen llaves aunque sólo contengan una sola sentencia, de esta forma se evita la introducción accidental de errores si se añaden posteriormente sentencias. Ejemplos:

```
if ($parametro == "DM") {  
    $dimension = array_sum($this->arrayPromedios) / 5;  
}
```

Asignación de nombres:

Cada elemento se nombra con una serie de reglas determinadas:

- Clases: los nombres de las clases comienzan con letra minúscula. Si la clase tiene un nombre compuesto por dos palabras, estas se separan con el siguiente caracter: “_”.
- Métodos: si el nombre del método está compuesto por dos palabras, la primera se escribe en minúscula y la segunda en mayúscula, una a continuación de la otra, sin espacios ni caracteres para separarlas. En caso de que el nombre del método esté formado por una sola palabra, entonces se escribe en minúscula.
- Variables: los nombres de las variables se escriben en minúscula, a excepción de aquellas variables que hacen referencia a abreviaturas relacionadas con el negocio en cuestión.

Tareas de Ingeniería

Las Tareas de Ingeniería se usan para describir las tareas que se realizan sobre el sistema. Estas tareas tienen relación con una Historia de Usuario, se especifica la fecha de inicio y fin de la tarea, se nombra al programador responsable de cumplirla y se describe qué se deberá hacer en dicha tarea.

Al concluir la asignación de tareas por historias de usuario se determinan 28 tareas de ingeniería a realizar. A continuación se detalla la tarea de ingeniería que se debe ejecutar para la Historia de Usuario # 2 que tiene prioridad alta en el negocio, el resto se pueden encontrar en el expediente de proyecto y en los [anexos](#).

Tarea	
Número de tarea: 4	Nombre Historias de usuario: Adicionar usuario
Nombre de la tarea: Permitir la adición de usuarios en el sistema.	
Tipo de tarea: Desarrollo (Desarrollo/Corrección/Mejora)	Puntos Estimados(días): 1
Fecha inicio: 13/03/2013	Fecha fin: 14/03/2013
Programador responsable: Yaima Morales Labrada	
Descripción: En esta tarea se implementa el código que permitirá que el administrador pueda adicionar usuarios.	

Tabla # 6: Tarea de Ingeniería

2.6 Conclusiones del capítulo

Luego de finalizar el capítulo, se arriba a las siguientes conclusiones:

- La descripción del escenario actual del proceso de evaluación de la satisfacción del cliente interno en el Centro fue necesario para la definición de la estructura del sistema.
- Se lograron definir 17 requisitos funcionales y una Historia de Usuario por cada requisito funcional establecido, además se definieron 20 requisitos no funcionales lo que dio lugar a la propuesta de solución.
- Se determinó la realización de 28 tareas de ingeniería lo que permitió establecer las actividades que se deben llevar a cabo en la implementación.

Capítulo 3: Validación de la propuesta de solución.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

Introducción

En este último capítulo se pretende verificar y validar el correcto funcionamiento del sistema, la verificación se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica y en la validación se ejecutan un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente.

Para la comprobación final del sistema ESCISoft (Software para la Evaluación de la Satisfacción del Cliente Interno) se aplicarán técnicas de validación de requisitos y métricas de software, además se realizarán pruebas unitarias y pruebas de aceptación.

3.1 Aplicación de técnicas de validación de requisitos y utilización de métricas

Técnicas de validación de requisitos

La validación de los requisitos tiene como objetivo comprobar que estos son correctos. Esta fase debe realizarse o de lo contrario se corre el riesgo de implementar una mala especificación, con el costo que eso conlleva. Es muy importante asegurar la validez de los requisitos antes de comenzar el desarrollo del software. Para ello debe hacerse una comprobación de la correspondencia entre las descripciones iniciales y la definición de los requisitos realizada, para verificar que responden a lo que desea el usuario final. Para llevar a cabo este proceso, se aplicaron las siguientes técnicas de validación de requisitos:

- Revisiones de requisitos: los requisitos se analizan sistemáticamente por un revisor técnico.

Al poner en práctica estas revisiones, se detectaron en una primera iteración cinco problemas relacionados con una mala expresión de los requisitos y falta de información a la hora de definirlos. Al realizar la segunda iteración se comprobó que se habían resuelto los errores antes encontrados y que la definición efectuada era correcta.

- Construcción de prototipos: el prototipado de interfaz de usuario es una técnica de representación aproximada de la interfaz de usuario de un software que permite a clientes y usuarios entender fácilmente la propuesta de los ingenieros de requisitos para resolver sus problemas de negocio.

Esta técnica se aplicó diseñando prototipos de interfaz de usuario para que el cliente entendiera de qué forma iba a quedar el sistema que se comenzaba a desarrollar, para ello se realizaron reuniones para mostrar los prototipos, en una primera reunión, el cliente no estuvo de acuerdo con tres de los prototipos realizados, luego se modificaron dichos prototipos y en la segunda reunión efectuada se consiguió la aprobación del cliente.

Métricas

Un elemento clave de cualquier proceso de ingeniería es la medición. Se emplean medidas para valorar la calidad de los sistemas que se construyen. Para conseguir esta medición se va a hacer uso de métricas para requisitos y para diseño.

Métrica para requisitos

Una vez definidos los requisitos del sistema, estos deben ser validados para asegurar que los requisitos que se determinaron especifican realmente el sistema que el cliente desea obtener.

Aplicación de la métrica Estabilidad de requisitos:

$$ETR = \left[\frac{RT - RM}{RT} \right] * 100$$

(Pressman, 2008)

Donde:

RT: Total de requisitos definidos.

RM: Cantidad de requisitos modificados.

Se identificaron un total de 17 requisitos funcionales, de los cuales 2 fueron modificados, por tanto:

$$ETR = [(17 - 2) / 17] * 100 = 88$$

Como resultado se obtuvo un valor de 88, este número demuestra que no se han realizado cambios significativos sobre los requisitos, son estables y por tanto, es confiable el diseño efectuado sobre ellos.

Aplicación de la métrica Especificidad de los requisitos:

La Especificidad de los requisitos se calcula como:

$$Q_1 = \frac{n_{ui}}{n_r} \quad (\text{Pressman, 2008})$$

Donde:

Q1: Especificidad de los requisitos.

nr: Total de requisitos definidos.

nui: Total de requisitos donde los revisores tuvieron interpretaciones idénticas.

El valor de esta métrica debe estar siempre entre 0 y 1. Mientras más cerca de 1 esté el valor de Q1, mayor será la consistencia de la interpretación de los revisores para cada requisito y menor será la ambigüedad en su especificación.

Se obtiene como resultado:

$$Q_1 = 17/17 = 1$$

Como el valor que se obtuvo de Q1 es 1, se puede interpretar que la especificación de los requisitos no posee ambigüedad. Esta característica garantiza mayor calidad en el proceso de especificación.

Aplicación de la métrica Grado de Validación de los requisitos:

El Grado de Validación de los requisitos mide la corrección en la definición de dichos requisitos. Este valor se calcula con la siguiente fórmula:

$$Q_3 = \frac{n_c}{(n_c + n_{nv})} \quad (\text{Pressman, 2008})$$

Donde:

Q3: Grado de Validación de los requisitos.

nc: Total de requisitos validados correctamente.

nnv: Total de requisitos no validados.

El resultado de esta métrica está siempre entre 0 y 1. El valor óptimo, es el más cercano a 1 e indica un alto nivel de corrección en la definición de los requisitos.

Se obtiene como resultado:

$$Q3 = 17/(17+0) = 17/17 = 1$$

La aplicación de esta métrica dio como resultado 1, por lo tanto se concluye que la definición de los requisitos es correcta.

Métricas para diseño.

Para medir el diseño se utilizarán las siguientes métricas orientadas a clases: Tamaño Operacional de Clase (TOC) y Acoplamiento entre Clases (AC).

Aplicación de la Métrica Tamaño Operacional de Clases:

A continuación se muestran los resultados obtenidos tras la aplicación de la métrica TOC, donde:

CA: cantidad de atributos de la clase. R: responsabilidad de la clase.

CO: cantidad de operaciones de la clase. CI: complejidad de implementación de la clase.

U: umbral. Reu: reutilización de la clase.

Clasificación	Valores de los umbrales
Pequeño	≤ 20
Medio	$20 < \leq 30$
Grande	> 30

Tabla #7: Clasificación de las clases según el tamaño

Atributo	Categoría	Criterio
Responsabilidad	Baja	Umbral \leq Promedio
	Media	Promedio \leq Umbral $\leq 2 * Promedio$

	Alta	Umbral > 2* Promedio
Complejidad de Implementación	Baja	Umbral <= Promedio
	Media	Promedio <= Umbral <= 2* Promedio
	Alta	Umbral > 2* Promedio
Reutilización	Baja	Umbral > 2* Promedio
	Media	Promedio <= Umbral <= 2* Promedio
	Alta	Umbral <= Promedio

Tabla #8: Rango de valores para la evaluación de los atributos relacionados con la métrica TOC

No	Clases	CA	CO	U	TC	R	CI	Reu
1	Participante	21	33	54	Grande	Media	Media	Media
2	Usuario	13	18	31	Grande	Baja	Baja	Alta
3	Encuesta	20	15	35	Grande	Baja	Baja	Alta
4	Pregunta	10	16	26	Medio	Baja	Baja	Alta
5	Respuesta	5	14	19	Pequeño	Baja	Baja	Alta
6	Grupo	9	12	21	Medio	Baja	Baja	Alta
7	Autenticación	19	16	35	Grande	Baja	Baja	Alta
8	AccionesUsuario	23	19	42	Grande	Media	Media	Media
9	AdministracionEncuestas	32	37	69	Grande	Media	Media	Media
10	Estadísticas	17	11	28	Medio	Baja	Baja	Alta

Tabla #9: Resultados de la aplicación de la métrica TOC

Se trabajó con un total de 10 clases para un promedio de umbral de 36, obteniéndose como resultados los datos que a continuación se muestran:

Cantidad de clases: 10	Baja	Media	Alta
Responsabilidad	70 %	30 %	0 %
Complejidad de implementación	70 %	30 %	0 %
Reutilización	0 %	30 %	70 %

Tabla #10: Resultados generales de la aplicación de la métrica TOC

Luego de aplicar la métrica TOC y de alcanzar los resultados anteriores se llega a la conclusión de que las clases que forman parte del sistema están dentro de la categoría Baja para los atributos responsabilidad y complejidad de implementación con un 70 % del total y además se determina que existe una alta reutilización en el diseño propuesto. Por lo tanto, se concluye que los resultados obtenidos según esta métrica son positivos.

Aplicación de la Métrica Relaciones entre Clases (RC):

A continuación se muestran los resultados obtenidos tras la aplicación de la métrica RC, donde:

CR: cantidad de relaciones de uso. CM: complejidad de mantenimiento.

A: acoplamiento entre clases. CP: cantidad de pruebas.

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1

	Medio	2
	Alto	Cantidad de relaciones de uso > 2
Complejidad del mantenimiento	Baja	Cantidad de relaciones de uso <= Promedio
	Media	Promedio <= Cantidad de relaciones de uso <= 2* Promedio
	Alta	Cantidad de relaciones de uso > 2* Promedio
Cantidad de pruebas	Baja	Cantidad de relaciones de uso <= Promedio
	Media	Promedio <= Cantidad de relaciones de uso <= 2* Promedio
	Alta	Cantidad de relaciones de uso > 2* Promedio

Tabla #11: Rango de valores para la evaluación de los atributos relacionados con la métrica

No	Clases	CR	A	CM	CP
1	Participante	1	Bajo	Baja	Baja
2	Usuario	3	Alto	Media	Media
3	Encuesta	4	Alto	Alta	Alta

4	Pregunta	2	Medio	Media	Media
5	Respuesta	1	Bajo	Baja	Baja
6	Grupo	2	Medio	Media	Media
7	Autenticación	1	Bajo	Baja	Baja
8	AccionesUsuario	1	Bajo	Baja	Baja
9	AdministracionEncuestas	1	Bajo	Baja	Baja
10	Estadísticas	1	Bajo	Baja	Baja

Tabla #12: Resultados de la aplicación de la métrica RC

Se obtuvo un promedio de 1.7 asociaciones entre clases. A continuación se muestran los resultados tras la aplicación de la métrica RC:

Cantidad de clases: 10	Baja	Media	Alta
Acoplamiento	60 %	20 %	20%
Complejidad de mantenimiento	60 %	30 %	10 %
Cantidad de pruebas	60 %	30 %	10 %

Tabla #13: Resultados generales de la aplicación de la métrica RC

Los resultados obtenidos tras la aplicación de la métrica RC demuestran que las clases del diseño poseen un bajo acoplamiento, ya que para este atributo la clasificación baja sumó un 60% del total. Los atributos complejidad de mantenimiento y cantidad de pruebas, sumaron un 90% en las categorías baja y media, lo que demuestra que no es necesario un elevado esfuerzo a la hora de realizar cambios, correcciones y pruebas al software.

3.2 Verificación del sistema

En la etapa de verificación se comprueba que el software cumple los requisitos funcionales y no funcionales de su especificación.

Pruebas unitarias

Las pruebas unitarias son aplicadas para verificar que el software cumple los requisitos funcionales y no funcionales de su especificación y también son empleadas para asegurar la calidad del código entregado. Además, son la mejor forma de detectar fallas tempranamente en el desarrollo y está demostrado que mientras más pronto se encuentren los errores, menos costará corregirlos. Para llevar a cabo las pruebas unitarias se decidió aplicar los métodos de pruebas de caja blanca y de caja negra.

❖ Pruebas de caja negra

Las pruebas de caja negra no consideran la codificación dentro de sus parámetros a evaluar, solo se enfocan en los requisitos establecidos y en la funcionalidad de la aplicación desarrollada. Para realizar estas pruebas se introducen las entradas en los diferentes campos del sistema y se examinan las salidas correspondientes. Si las salidas no son las previstas, entonces la prueba ha detectado un fallo en el software.

Para aplicar el método de pruebas de caja negra se va a hacer uso de la técnica de partición equivalente mediante el diseño de casos de prueba, esta técnica se basa en la división del campo de entrada en un conjunto de clases de equivalencia, estas clases representan un grupo de datos de entrada que definen estados válidos y no válidos del sistema y se obtienen a partir de los casos de prueba.

Luego de aplicar las pruebas de caja negra se obtuvieron los siguientes resultados:

Iteraciones	Iteración 1	Iteración 2
No conformidades	11	0

Tabla #14: Resultados de las pruebas de caja negra

Las no conformidades encontradas en la primera iteración estaban relacionadas con la validación de los datos que se introducen en el sistema, pudiendo comprobar en la segunda iteración realizada que estos errores fueron corregidos y que la aplicación desarrollada funciona correctamente.

Los resultados alcanzados tras la aplicación de las pruebas de caja negra se pueden encontrar en el Expediente de Proyecto, específicamente en la carpeta Casos de prueba basados en Historias de Usuario. Para ver un caso de prueba aplicado, consultar [anexos](#).

❖ Pruebas de caja blanca

Las pruebas de caja blanca se basan en el conocimiento de la lógica interna del código del sistema, contemplan los distintos caminos que se pueden generar teniendo en cuenta las estructuras condicionales o los distintos estados del mismo.

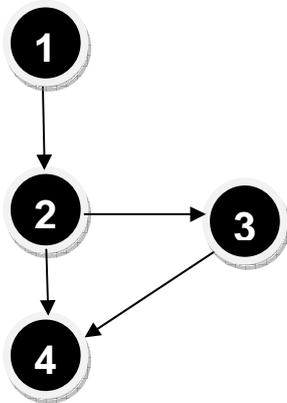
Para poner en práctica este tipo de pruebas se procede a utilizar la técnica del camino básico que permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de una funcionalidad y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Para la obtención de la complejidad ciclomática se emplea una representación del flujo de control en forma de grafo.

La complejidad ciclomática es una medida que aporta una valoración cuantitativa de la complejidad lógica de un programa. Dentro del contexto de la técnica del camino básico, representa el número de caminos independientes de un programa.

A continuación se aplica la prueba de caja blanca a la funcionalidad Adicionar usuario utilizando el método del camino básico, para ello se procede a enumerar las sentencias del código y a partir del mismo se construye el grafo de flujo asociado.

```
public static function adicionarUsuario($new_user, $new_pass,$new_full_name,$parent_user,$new_email)
{
    $result = false;                                (1)
    oUser = new self;                               (1)
    $oUser->users_name = $new_user;                 (1)
    $oUser->password = hash('sha256', $new_pass);   (1)
    $oUser->full_name = $new_full_name;            (1)
    $oUser->parent_id = $parent_user;              (1)
    $oUser->lang = 'auto';                          (1)
    $oUser->email = $new_email;                    (1)
    if ($oUser->save())                              (2)
    {
        $result = $oUser->uid;                       (3)
    }
    return $result;                                (4)
}
```

Código fuente de la funcionalidad Adicionar usuario



Grafo de flujo asociado a la funcionalidad Adicionar usuario

Luego de haber realizado la construcción del grafo de flujo se procede a calcular la complejidad ciclomática mediante tres fórmulas que se describen a continuación, las cuales deben exponer el mismo resultado para asegurar que el cálculo de la complejidad es correcto.

Para cada fórmula $V(G)$ representa el valor del cálculo.

1. $V(G) = (A - N) + 2$ (Pressman, 2008)

Donde A es el número de aristas y N es el número de nodos contenidos en el grafo.

$$V(G) = (4 - 4) + 2$$

$$V(G) = 2$$

2. $V(G) = P + 1$ (Pressman, 2008)

Donde P es el número de nodos predicados contenidos en el grafo. Los nodos predicados son aquellos que parten de dos o más aristas.

$$V(G) = 1 + 1$$

$$V(G) = 2$$

3. $V(G) = R$ (Pressman, 2008)

Donde R es la cantidad total de regiones.

$$V(G) = 2$$

El número de regiones del grafo es igual a la complejidad ciclomática.

El cálculo efectuado anteriormente dio como resultado el mismo valor en todos los casos, la complejidad ciclomática es de 2, este valor indica que existen 2 posibles caminos por donde el flujo puede circular y determina el número de casos de prueba que se deben realizar para asegurar que se ejecute cada

sentencia al menos una vez. A continuación se representan los caminos básicos por los que puede transitar el flujo:

Camino básico #1: 1-2-3-4

Camino básico #2: 1-2-4

Luego de haber determinado los caminos, se le realiza un caso de prueba a cada uno.

Caso de prueba para el Camino Básico #1:

Descripción: los datos de entrada cumplirán con los siguientes requisitos: los campos nombre de usuario y nombre completo aceptan cualquier combinación de las letras del alfabeto, ya sea en minúsculas o en mayúsculas (a-z y A-Z) y además, admite introducir números enteros. El campo correo electrónico debe tener el formato aaaa@bbb.ccc.

Condiciones de ejecución:

- El sistema debe estar instalado y ejecutándose correctamente.
- El usuario debe estar autenticado con los permisos necesarios.

Entrada: \$new_user = nuevo, \$new_email = usuario@uci.cu, \$new_full_name = nuevo1

Resultados esperados: Se realiza la adición de un nuevo usuario satisfactoriamente.

Caso de prueba para el Camino Básico #2:

Descripción: los datos de entrada cumplirán con los siguientes requisitos: los campos nombre de usuario y nombre completo aceptan cualquier combinación de las letras del alfabeto, ya sea en minúsculas o en mayúsculas (a-z y A-Z) y además, además admite introducir números enteros. El campo correo electrónico debe tener el formato aaaa@bbb.ccc.

Condiciones de ejecución:

- El sistema debe estar instalado y ejecutándose correctamente.
- El usuario debe estar autenticado con los permisos necesarios.

Entrada: \$new_user = nuevo, \$new_email = usuario@uci, \$new_full_name = nuevo1

Resultados esperados: no se realiza la adición de un nuevo usuario porque se introducen datos inválidos en el campo correo electrónico.

Con la aplicación de un caso de prueba para cada camino básico encontrado se garantiza que se ejecute cada sentencia al menos una vez. Al concluir la ejecución de estas pruebas se obtuvieron los resultados esperados en cada caso de prueba realizado.

3.3 Validación de sistema

Con la validación se pretende comprobar que el software cumple las expectativas que el cliente espera. Para llevar a cabo la validación del sistema se decidió aplicar pruebas de aceptación por cada Historia de Usuario definida.

Pruebas de aceptación

Las pruebas de aceptación son destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente final. Estas pruebas aseguran el comportamiento del sistema y especifican los aspectos a probar cuando una Historia de Usuario ha sido correctamente implementada.

A continuación se muestra el caso de prueba de aceptación para una Historia de Usuario de prioridad alta en el negocio, el resto se puede encontrar en el expediente de proyecto y en los [anexos](#).

Caso de prueba de aceptación para la Historia de Usuario # 2

Caso de prueba de aceptación	
Código: HU2_P1	Historia de Usuario : 2
Nombre: Adicionar usuario	
Descripción: Se debe validar que en el sistema sea posible adicionar usuarios y que para hacerlo se llenen los campos nombre de usuario y correo electrónico de manera obligatoria. Además, se debe verificar que no se introduzcan datos incorrectos ni se dejen campos obligatorios vacíos.	
Entrada/Pasos de ejecución: se realiza la validación de la adición de usuarios en el sistema introduciendo datos incorrectos y dejando campos vacíos.	
Resultados esperados:	

Se ha validado correctamente la adición de usuarios en el sistema.
Evaluación de la prueba: Prueba satisfactoria.

Tabla #15: Caso de prueba de aceptación

Resultados:

Iteraciones	Iteración 1	Iteración 2
Evaluación de la prueba	3 pruebas insatisfactorias	0 pruebas insatisfactorias

Tabla #16: Resultados de las pruebas de aceptación

Para la aplicación de este tipo de pruebas, se realizaron dos iteraciones, en la primera se obtuvieron 3 pruebas insatisfactorias relacionadas con la interfaz de usuario y en la segunda iteración se logró finalmente la aceptación del cliente. Para ver el acta de aceptación, consultar [anexos](#).

3.4 Validación de las variables

Con el desarrollo del sistema, se facilita el proceso de identificación de causas de insatisfacción de los clientes internos para el apoyo a la toma de decisiones administrativas en el Centro CEGEL, esto se evidencia con los siguientes elementos:

- Anteriormente, en el Centro CEGEL se aplicaban encuestas de manera manual para evaluar la satisfacción de los clientes internos y en algunos casos se perdía la información recogida debido a que las respuestas se encontraban en formato duro. Otro problema que afectaba al Centro es que resultaba muy difícil realizar manualmente el análisis de los datos pues la revisión de las respuestas requería de mucho tiempo, los elementos antes mencionados dificultaban la identificación de las causas que le producen insatisfacción a los clientes internos del Centro.
- Hoy en día, se pueden aplicar las encuestas para evaluar la satisfacción de los clientes internos del Centro de forma sencilla, con solo acceder al sistema ESCISoft se podrá responder cada encuesta que haya sido publicada por el Asesor de Calidad, lográndose la informatización del proceso de

evaluación y evitando la pérdida de la información recopilada pues las respuestas quedan almacenadas en la aplicación. Además, el sistema desarrollado permite a la Dirección del Centro y al Asesor de la Calidad consultar los reportes con los resultados, donde se muestran gráficas con el Índice de Satisfacción del Cliente Interno de cada elemento que fue evaluado, facilitándose de este modo el análisis de los datos y logrando una inmediatez en los resultados, pudiendo identificar de manera rápida y sencilla las causas que producen insatisfacción a los clientes internos del Centro CEGEL.

En el año 2010 se llevó a cabo una investigación con el título "*Evaluación de la satisfacción de los clientes internos en el proceso de desarrollo de software de la Facultad 15 de la Universidad de las Ciencias Informáticas*" donde se evaluó por primera vez el Índice de Satisfacción del Cliente Interno mediante la aplicación de encuestas. Se utilizó una muestra de 34 clientes internos de los proyectos existentes en la Facultad 15 (actualmente Facultad 3), para ello se imprimieron 68 encuestas, dos por cada cliente. El proceso de recepción de las respuestas demoró 15 días y el análisis de los resultados, una semana, lo que demuestra la dificultad y la demora que existía para identificar las causas de insatisfacción de los clientes internos de la Facultad.

Para comprobar las facilidades que brinda la herramienta desarrollada, se aplicó una prueba piloto a la misma cantidad de clientes internos que se tomó como muestra en la investigación desarrollada en el año 2010, donde se midió la satisfacción de todos los roles definidos para el proceso de evaluación. Se evaluaron 6 analistas, 2 administradores de la configuración, 9 desarrolladores, 7 diseñadores de base de datos y 10 revisores técnicos pertenecientes a los proyectos Fiscalía, Tribunales, Cámara de Comercio y Calidad. Se arribó a los siguientes resultados: la recepción de las respuestas se obtienen en un promedio de 20 minutos y el análisis de los resultados en 10 minutos, este análisis comienza desde que se generan las estadísticas de la encuesta hasta que se identifican las causas de insatisfacción, de esta manera se demostró el cumplimiento del problema de investigación planteado, pudiendo constatar que con el sistema creado se logra una inmediatez en los resultados, se facilita la identificación de las causas que producen insatisfacción a los clientes internos del Centro CEGEL y se mejora el proceso de evaluación de la satisfacción. Para ver los resultados de la prueba piloto realizada, ver los [Anexos](#).

3.5 Conclusiones parciales

A lo largo del capítulo se elaboraron y analizaron los elementos fundamentales que forman parte de la verificación y validación del sistema, así como la validación de la investigación, obteniendo los siguientes resultados:

- Se emplearon Técnicas de Validación de requisitos lo que permitió asegurar la validez de los requisitos antes de comenzar el desarrollo del software.
- Se aplicaron métricas para requisitos obteniendo como resultado que los requisitos definidos son estables y por tanto, es confiable el diseño efectuado sobre ellos, además se concluyó que la definición realizada es correcta ya que el 100% de dichos requisitos no poseen ambigüedad.
- También se aplicaron métricas para diseño, en el caso de la métrica RC, los resultados demuestran que las clases del diseño poseen un bajo acoplamiento y tras la aplicación de la métrica TOC se llegó a la conclusión de que existe una baja responsabilidad y complejidad de implementación y que existe una alta reutilización en el diseño propuesto con un valor del 70% del total. Por lo tanto, se concluye que los resultados obtenidos según estas métricas son positivos.
- Se aplicaron pruebas unitarias utilizando los métodos de prueba de caja blanca y de caja negra lo que permitió verificar que el software cumple los requisitos funcionales y no funcionales de su especificación.
- La aplicación de las pruebas de aceptación garantizó que el sistema desarrollado funciona correctamente y permitió verificar la aceptación del sistema por parte del cliente.
- Por último, se validaron las variables que forman parte del problema de la investigación, demostrando que con el sistema que se desarrolló, se facilita la identificación de las causas que producen insatisfacción a los clientes internos del Centro CEGEL y se mejora el proceso de evaluación de la satisfacción.

Conclusiones Generales y Recomendaciones

CONCLUSIONES GENERALES

Luego de realizar la investigación se obtuvieron las siguientes conclusiones:

- Los clientes internos juegan un papel imprescindible en la producción de software debido a que el resultado de su trabajo contribuye significativamente al logro de un producto final con calidad.
- Cuando se habla de satisfacción del cliente no se refiere solamente al hecho de complacer a los clientes externos, sino también a los clientes internos, es por ello que se debe prestar especial atención a esas personas cuyo trabajo garantiza que se logren los objetivos de una organización.
- A partir del estudio de las herramientas seleccionadas para diseño, entorno de desarrollo y gestión de datos se desarrolló una aplicación web para evaluar la satisfacción de los clientes internos en el proceso de desarrollo de software del Centro de Gobierno Electrónico mediante la realización de encuestas.
- Con la herramienta creada se pueden identificar de manera fácil y sencilla las causas de insatisfacción de los clientes internos luego de aplicar las encuestas, permitiendo que la dirección del Centro CEGEL tome medidas encaminadas a mejorar los aspectos donde se encuentran insatisfechos los clientes internos.

Recomendaciones

La realización de la investigación permitió desarrollar una herramienta para evaluar la satisfacción de los clientes internos en el proceso de desarrollo de software de los proyectos del Centro CEGEL, aunque se recomienda:

1. Añadir funcionalidades para mostrar el Índice de Satisfacción del Cliente Interno a nivel de Centro y a nivel de proyecto.
2. Agregarle a la herramienta desarrollada funcionalidades que permitan generar diferentes alternativas o escenarios de decisión para mejorar los aspectos donde se encuentran insatisfechos los clientes internos del Centro como apoyo a la toma de decisiones.

Referencias bibliográficas

REFERENCIAS BIBLIOGRÁFICAS

1. (IEEE), Instituto de Ingenieros Eléctricos y Electrónicos. 1990. IEEE 610-1990.
2. Pressman, Roger S. 1998. Ingeniería de Software. Un enfoque práctico.
3. Fundación Vasca para la Excelencia. 2011. Calidad Total: Principios y Modelos de gestión. Certificación ISO. Satisfacción del Cliente interno y externo.
4. Vivó, Ana Rosa Sánchez. 2012. Procedimiento para medir la satisfacción del cliente externo. Tesis de Maestría. Instituto Superior Politécnico José Antonio Echeverría (CUJAE).
5. Pereiro, Jorge. 2008. La satisfacción del cliente en ISO 9001.
6. Acevedo, J. A, S. y o. 1995. El modelo de organización de la Empresa Industrial. Ciudad del La Habana.
7. Albrecht, Karl. 1992. "SERVICIO AL CLIENTE INTERNO".
8. IBM (International Business Machines). 2008. La nueva colaboración: facilitar la innovación, transformar el lugar de trabajo.
9. Barón, Ruth Alicia Cadena. Olarte, Lisbed Giraldo. Mora, Sandra Marcela. 2010. Evaluación de la satisfacción del cliente interno en la IPS Oral Medic S.A.
10. Acuña, Uriel Ladrón de Guevara. 2009. Medición de la satisfacción del cliente interno en una empresa de transformación. Universidad de Veracruz.
11. López, Maritza Rodríguez. Ruiz, José Rafael Laborí. 2010. Satisfacción por los servicios brindados en la sala de Fisioterapia del Hospital General "Dr. Juan Bruno Zayas Alfonso" de Santiago de Cuba.
12. Cruz, Ing. Aimée Hernández. Jerez, Ing. María del Carmen Grillo. Álvarez, Héctor Juan Díaz. 2002. ¿Qué hacer por la calidad?
13. Sistema Integrado de Gestión. 2009. Procedimiento para medir la satisfacción de los funcionarios de la Universidad frente a los servicios internos de los procesos. Universidad de Caldas.
14. Blanco, Rubén González. Tobalina, Sergio Pérez. 2012. LESE-2 Introducción a Rational Rose. Universidad Politécnica de Cataluña.
15. Giraldo, Luis. Zapata, Yuliana. 2005. Herramientas de desarrollo de Ingeniería de Software para Linux.

16. LimeSurvey. Características. En línea: <http://www.limesurvey.org/es/acerca-de-limesurvey/caracteristicas>
17. Sommerville, Ian. 2005. Ingeniería de Software. Séptima edición.
18. Muñoz, Coral Calero. 2006. Métricas del software: conceptos básicos, definición y formalización.
19. Fornaris, Maité Sánchez. Rabí, Dayanis Alcántara. 2010. Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica.
20. Rodríguez, Jorge. 2006. Pruebas unitarias.
21. Departamento de Informática. Tema 7: Validación. Universidad de Valladolid.
22. Noa, Ada Milagro Guilarte. Rodríguez, Yennis Guerra. 2011. Definición de indicadores para medir la satisfacción de los clientes internos en el proceso de desarrollo de software de los proyectos del Centro de Gobierno Electrónico de la Facultad 3. Universidad de las Ciencias Informáticas.
23. Ochoa, Sergio. 2005. Introducción a los Patrones (Diseño y Arquitectura).
24. Gómez, José Jorge Márquez. 2011. ARQUITECTURA MVC. VISIÓN GENERAL.
25. Gamma. 1995. DesingPatterns.

Bibliografía consultada

1. Priolo. S. Programación Extrema.
2. Una introducción a Apache. En línea: http://linux.ciberaula.com/articulo/linux_apache_intro
3. Veloso. P. Uso de patrones de arquitectura.
4. Pimentel. E. Patrones de Diseño. Universidad de Málaga.
5. Reynoso. B. C. Métodos Ágiles en Desarrollo de Software. Introducción a la Arquitectura de Software. Universidad de Buenos Aires.
6. Hernández, Denia Madruga. Mazquiarán, Isabel de la Concepción Morales. 2010. Evaluación de la satisfacción de los clientes internos en el proceso de desarrollo de software de la Facultad 15 de la Universidad de las Ciencias Informáticas.
7. Marugan, Axel. 2013. Tim Cook recibe un 93 % de satisfacción de sus empleados. En línea: <http://applesencia.com/2013/04/tim-cook-recive-93-satisfaccion-empleados>
8. En línea: <http://www.wisis.ufg.edu.sv/www.wisis/documentos/TE/338.476%2077-M379d/338.476%2077-M379d-CAPITULO%20IV.pdf>
9. <http://loogic.com/65-millones-de-inversion-en-glassdoor/>