Universidad de las Ciencias Informáticas
Facultad 4

Aplicación Web para la Mejora en la Gestión de Servicios de Transconsul S.A.

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

Autor: Técn. Carlos Brayan Rámila Chorens

Tutora: MSc. Yadira Ramírez Rodríguez

La Habana, 2024

La eficaz gestión de citas y servicios legales no solo es una necesidad operativa, sino un pilar fundamental para garantizar la confianza y satisfacción de nuestros clientes en cada interacción.
John Smith, Experto en Gestión Legal

Dedicatoria

A mi familia, amigos, compañeros y profesores, quienes con su amor, apoyo, sabiduría y aliento han sido mi mayor inspiración y motivación en cada paso del camino. Este trabajo está dedicado a ustedes, quienes han hecho posible este logro con su constante presencia y ayuda incondicional.

Agradecimientos

Quiero expresar mi más profundo agradecimiento a todas las personas que han sido parte de este viaje académico. A mi familia, por su amor incondicional y su constante apoyo en cada desafío que he enfrentado. A mis amigos, por estar siempre presentes, brindándome su amistad y aliento en los momentos difíciles. A mis compañeros, por compartir este camino conmigo, por los momentos de colaboración y aprendizaje mutuo. Y a mis profesores, por su orientación, sabiduría y dedicación, que han sido fundamentales para mi crecimiento académico y personal.

En especial, quiero agradecer a mi tutora y profesora, M.Sc. Yadira Ramírez Rodríguez, por su invaluable guía y apoyo durante todo el proceso de investigación y redacción de esta tesis. Su paciencia, experiencia y compromiso han sido fundamentales para alcanzar este logro.

Por último, pero no menos importante, quiero agradecerme a mí mismo por mi perseverancia, determinación y dedicación para alcanzar mis metas. Sin mi propia fuerza interior y convicción, este logro no habría sido posible.

Declaración de autoría

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

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

Técn. Carlos Brayan Rámila Chorens	MSc. Yadira Ramírez Rodríguez
Autor	Tutora

Resumen

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

Palabras clave: word1, word2, word3.

Índice general

Índice de figuras

Índice de tablas

Lista de algoritmos

Lista de códigos fuentes

Introducción

El desarrollo vertiginoso de las nuevas tecnologías, particularmente en las ramas de la informática y las telecomunicaciones, evidencia que es esta la era con mayor evolución científico-técnica de todos los tiempos. Este desarrollo tecnológico acelerado incide sustancialmente en el mundo referencial del ser humano, pues, si bien facilita la adquisición y almacenamiento de nuevos conocimientos, nuevos enfoques y/o perspectivas y acciones que ayer mismo parecían inaccesibles pero, de la misma manera, le están condicionando y obligando a adaptaciones y replanteamientos en todos los órdenes de su existencia.

En el campo del desarrollo de software también se constata un avance sin precedentes a nivel mundial, lo cual posibilita la construcción de aplicaciones y sistemas informáticos capaces de solventar la amplia amalgama de problemas teóricos y prácticos de la sociedad. Diversas áreas de la vida diaria se ven afectadas continuamente por la industria del software, tales como los aeropuertos, los bancos, las grandes empresas, las instituciones policiales y militares entre otras.

Los sistemas de gestión son una rama importante dentro de la industria del desarrollo de software puesto que resuelven un gran número de situaciones a través de la automatización de procesos, lo cual facilita el trabajo en oficinas, elimina el tiempo de respuesta por parte de los proveedores de servicios a la población, suprime engorrosos bloques de papeles, propicia el trabajo de los seres humanos con grandes volúmenes de información, el control y manipulación de estadísticas y una innumerable cantidad de beneficios y posibilidades más, que a la postre, repercuten de forma positiva en la economía y la calidad de vida de todos aquellos países que explotan este tipo de sistemas informáticos.

En Cuba, con la intención de utilizar la amplia gama de posibilidades que brinda la industria del software se trabaja hace ya algún tiempo en el desarrollo de este tipo de herramientas, donde la Universidad de las Ciencias Informáticas asume un rol protagónico en esta tarea. Justamente, la informatización de los procesos que se llevan a cabo en una empresa o entidad resulta un elemento clave. Un proceso que resulta prioritario automatizar por la importancia que tiene en la sociedad se encuentra relacionado con la gestión de citas pues de este dependerá el eficiente control y mejor planificación del tiempo en las oficinas de trámites.

La sociedad civil de servicios Transconsul Sociedad Anónima (S.A.), constituida en el mes de abril de 2020, es una organización que ofrece servicios legales a personas naturales y jurídicas en términos de

asesoría, asistencia, representación y legalización de documentos. Asimismo, posee competencia legal para contratar sus servicios tanto a clientes nacionales como extranjeros. Dicha sociedad mercantil enfrenta diversos desafíos en la gestión de citas y comunicación, lo que ha provocado una percepción negativa por parte de los clientes. Esto se puede percibir en la valoración de 1.8, así como los comentarios negativos en su página de Facebook, la poca interacción de los usuarios respecto a las publicaciones en X, además de la ausencia de respuestas por parte de la organización en Google. Los clientes expresan frustración debido a la dificultad para agendar citas pues actualmente estas solo se pueden programar mediante llamadas telefónicas en un horario específico (lunes de 9:00 AM a 12:00 PM). En primer lugar, los números de teléfono ofrecidos para dicha actividad se corresponden con líneas fijas, de modo que eso entorpece aún más la capacidad de la persona para realizar su solicitud. En segundo lugar, se advierte una brecha en las respuestas que Transconsul S.A ofrece a consultas y quejas por la lentitud en la gestión de los servicios, tanto a través de sus redes sociales como en las comunicaciones vía telefónica. Esta situación actual proyecta una imagen negativa de la organización y también limita la eficiencia de la misma.

A partir de la situación antes descrita se plantea como problema de la investigación: ¿Cómo mejorar la gestión de los servicios de Transconsul S.A.?

Teniendo como objeto de estudio: Sistemas de gestión de servicios.

Se define como objetivo general: Desarrollar una aplicación web para la mejora en la gestión de servicios de Transconsul S.A. que garantice la satisfacción del cliente y mejore la imagen pública de la empresa; enmarcado en el campo de acción: El proceso de gestión de servicios de Transconsul S.A.

Para dar cumplimiento al objetivo planteado se trazan las siguientes tareas de investigación:

- Análisis de los referentes teóricos sobre mejores prácticas en sistemas de gestión de citas en línea, servicios y comunicación con los clientes.
- Identificación de las necesidades y requisitos específicos de Transconsul S.A. para la gestión de citas y servicios legales.
- Desarrollo de una aplicación web que incluya un sistema de gestión de citas y servicios.
- Implementación de la solución tecnológica para asegurar la satisfacción del cliente y renovar la imagen pública de la empresa.
- Realización de pruebas funcionales y de usabilidad para validar el correcto funcionamiento y la aceptación de la aplicación web por parte de los usuarios.

Para el desarrollo de la investigación se emplean los siguientes métodos científicos:

Métodos teóricos:

- Analítico-Sintético: para el estudio de la situación actual y la identificación de requisitos y necesidades.
- Modelación: para el diseño de la aplicación y la conceptualización de sus componentes y funcionalidades.

Métodos empíricos:

- Entrevistas: con empleados y clientes de Transconsul S.A. para recoger sus necesidades y expectativas.
- Observación: análisis de la interacción actual de los clientes con Transconsul S.A..
- Análisis de Redes Sociales: Evaluación de la retroalimentación de los clientes en las redes sociales de Transconsul S.A., identificando patrones comunes en las quejas y sugerencias para mejorar los servicios de la empresa.

CAPÍTULO 1

Fundamentación Teórica

1.1. Introducción al capítulo

En este capítulo se sintetiza la búsqueda y análisis de la información relacionada con el dominio del problema. Son descritos los conceptos fundamentales relacionados con la investigación y analizadas algunas soluciones existentes referentes al mismo entorno. Aborda además, la selección y descripción de las principales tecnologías y herramientas que serán utilizadas para llevar a cabo la implementación de la solución a desarrollar.

1.2. Sistema de gestión de citas y servicios

Tabla 1.1. Roles del sistema

Roles	Interacción en Medicando
Administrador	Configuración y mantenimiento del sistema de gestión de citas y servicios de legalización. Gestión de usuarios y asignación de roles y permisos. Misión del funcionamiento general del sistema y resolución de problemas técnicos. Actualización de información relevante, como horarios disponibles y servicios ofrecidos.
Cliente	Solicitar servicios legales y programar citas con abogados. Proporcionar la documentación necesaria para los procesos de legalización. Confirmar asistencia a las citas programadas y seguir las instrucciones proporcionadas por el bufete.
Continúa en la siguiente página	

Tabla 1.1. Continuación de la página anterior

Roles	Interacción en Medicando
Legalizadora	Coordinar y llevar a cabo los procesos de legalización de documentos ante las autoridades correspondientes. Verificar la autenticidad y validez de los documentos presentados para su legalización. Mantener registros precisos de los documentos procesados y su estado actual. Comunicarse con clientes y autoridades pertinentes para resolver cualquier problema o solicitud relacionada con la legalización.
Registradora	Gestionar los registros de citas programadas, canceladas y reprogramadas. Actualizar la disponibilidad de horarios en el sistema según las citas programadas. Coordinar con el personal administrativo para garantizar una facturación precisa de los servicios prestados. Proporcionar asistencia a los clientes en el proceso de programación de citas y en la resolución de problemas relacionados con la agenda.

1.3. Sistemas Homólogos

1.4. Metodología de desarrollo de software

Se siguieron los pasos que define la metodología Progrmación Extrema (XP) para guiar el proceso de desarrollo de software, atendiendo al esquema que se proponen en (**escribano2002introduccion**), adoptando sus 4 fases (Planificación, Diseño, Desarrollo y Prueba). XP es la más destacada de los procesos ágiles de desarrollo de software formulada por Knet Beck7. Tiene éxito porque hace hincapié en la satisfacción del cliente. Faculta a sus desarrolladores para responder con seguridad a las necesidades cambiantes de los clientes, incluso en etapas tardías del ciclo de vida del producto. Enfatiza el trabajo en equipo. Los jefes de proyecto, clientes y desarrolladores son socios iguales en un equipo de colaboración que se auto-organiza en torno al problema a resolver de la forma más eficiente posible. XP mejora un proyecto de software en cinco aspectos esenciales; la comunicación, la sencillez, la retroalimentación, el respeto y el valor. Los programadores extremos se comunican constantemente con sus clientes y colegas de trabajo, mantienen su diseño simple y limpio. Reciben retroalimentación probando su software a partir del primer día y entregan el sistema a los clientes tan pronto como sea posible e implementan cambios como se sugiere. Cada pequeño éxito profundiza su respeto por las contribuciones únicas de cada uno y cada miembro del equipo. Con esta base los programadores extremos son capaces de responder con valentía a las cambiantes necesidades y la tecnología (**joskowicz2008reglas**).

1.5. Herramientas y tecnología de desarrollo de software

La elección y uso de herramientas y tecnologías apropiadas es crucial para el éxito del desarrollo del sistema de gestión de citas y servicios de Transconsul S.A.. Al evaluar y seleccionar cuidadosamente estas

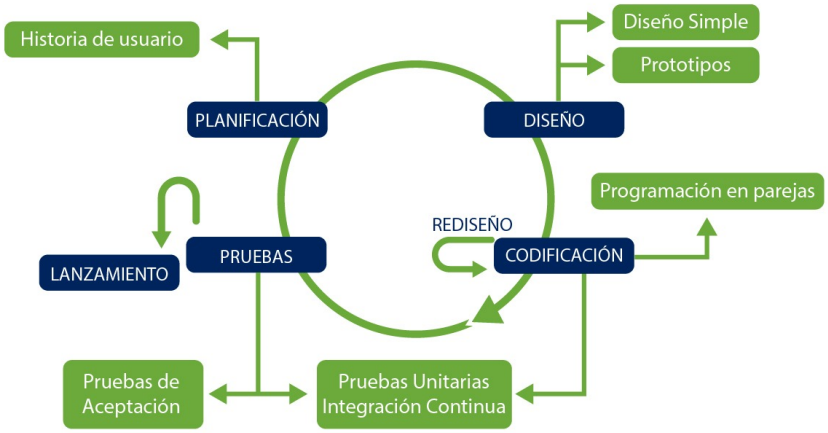

Figura 1.1. Pilares de la Metodología XP

herramientas, los programadores pueden trabajar de manera más eficiente y efectiva, lo que a su vez se traducirá en un producto final de alta calidad. Por lo tanto, es importante llevar a cabo una investigación exhaustiva de las herramientas y tecnologías disponibles, con el fin de identificar aquellas que mejor se adapten a las necesidades específicas del proyecto. De esta manera, se puede garantizar que el equipo de desarrollo tenga acceso a las herramientas y tecnologías necesarias para trabajar de manera efectiva y lograr los objetivos del proyecto de manera satisfactoria.

- Marco de trabajo: Con Django, puedes llevar aplicaciones web desde el concepto hasta el lanzamiento en cuestión de horas. Django se encarga de gran parte de las molestias del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto.
    - Completamente cargado: Django incluye docenas de extras que puedes usar para manejar tareas comunes de desarrollo web. Django se encarga de la autenticación de usuarios, la administración de contenidos, los mapas del sitio, las fuentes Sindicación Realmente Simple (RSS) y muchas más tareas, desde el primer momento.
    - Tranquilamente seguro: Django se toma en serio la seguridad y ayuda a los desarrolladores a evitar muchos errores de seguridad comunes, como la inyección Lenguaje de Consulta Estructurada (SQL), secuencias de comandos entre sitios, falsificación de solicitudes entre sitios y clickjacking. Su sistema de autenticación de usuarios proporciona una forma segura de administrar cuentas de usuarios y contraseñas.
    - Extremadamente escalable: Algunos de los sitios más concurridos del planeta utilizan la capacidad de Django para escalar de forma rápida y flexible para satisfacer las demandas de tráfico más intensas.

  - Increíblemente versátil: Empresas, organizaciones y gobiernos han utilizado Django para construir todo tipo de cosas, desde sistemas de gestión de contenidos hasta redes sociales y plataformas informáticas científicas.

- Entorno de desarrollo: Visual Studio Code es un editor de código fuente liviano pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C++, Java, C#, Python, PHP, Go, .NET)
  Un 'entorno' en Python es el contexto en el que se ejecuta un programa Python que consta de un intérprete y cualquier cantidad de paquetes instalados.

  El venv módulo admite la creación de .^{en}tornos virtuales"livianos, cada uno con su propio conjunto independiente de paquetes Python instalados en sus sitedirectorios. Un entorno virtual se crea sobre una instalación de Python existente, conocida como Python ?base? del entorno virtual, y opcionalmente puede aislarse de los paquetes en el entorno base, de modo que solo estén disponibles aquellos instalados explícitamente en el entorno virtual.

  Cuando se utilizan desde un entorno virtual, las herramientas de instalación comunes, como pip , instalarán paquetes de Python en un entorno virtual sin necesidad de que se le indique explícitamente que lo haga.
- Sistema Gestor de Base de Datos: se utilizó MySQL, este es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Es uno de los gestores más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación, es debida a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración (**daniel_pecos_martinez_postgresql_2024**). Las características fundamentales que refleja para su elección por encima de otros gestores de base de datos son: el coste gratuito, la velocidad operacional, facilidad de uso y de integración con la mayor parte de los entornos de programación, la existencia de una nutrida y activa comunidad (**gilfillan_biblia_2003**).

CAPÍTULO 2

Características y diseño del sistema

2.1. Introducción al capítulo

En el presente capítulo se presenta una propuesta de solución para los problemas detectados en el estudio del estado del arte realizado en el Capítulo 1. Se mencionan los roles que intervienen en la solución. También se definen las HU, la planificación de entrega de versiones del producto y el diseño del sistema de acuerdo a las fases que propone XP.

2.2. Propuesta de solución

Luego de haber analizado las necesidades del sistema y seleccionado las herramientas para la implementación, se definen los módulos a desarrollar para dar solución al problema planteado.

- Autenticación y Autorización: Este módulo se encargaría de gestionar la autenticación de usuarios, el registro de cuentas y la autorización de acceso a diferentes partes de la aplicación.
- Gestión de Citas: Este módulo sería el núcleo de la aplicación y se encargaría de permitir a los usuarios programar, modificar y cancelar citas. Debería incluir un calendario interactivo, notificaciones de citas programadas y recordatorios automáticos.
- Gestión de Clientes: Aquí se manejaría toda la información relacionada con los clientes, incluyendo su perfil, historial de citas, documentos legalizados y cualquier otra información relevante.
- Servicios Legales y Legalización: Este módulo se enfocaría en la gestión de los servicios legales ofrecidos por el bufete de abogados, así como en el proceso de legalización de documentos. Debería permitir a los usuarios solicitar servicios específicos, enviar documentos y realizar seguimiento del estado de sus trámites.
- Comunicación y Notificaciones: Este módulo sería responsable de facilitar la comunicación entre el bufete de abogados y sus clientes, a través de mensajes directos, correos electrónicos o notificaciones automáticas sobre el estado de las citas y servicios solicitados.

- Administración del Sistema: Este módulo sería utilizado por los administradores del sistema para gestionar usuarios, configurar ajustes de la aplicación, generar informes y realizar tareas de mantenimiento.
- Análisis y Reportes: Este módulo se encargaría de recopilar datos sobre el uso de la aplicación, el rendimiento del sistema y la satisfacción del cliente, para generar informes y análisis que ayuden a mejorar la eficiencia y calidad del servicio.

Para el desarrollo de los módulos propuestos se siguen los pasos que establece la metodología ágil XP, donde se respetará el esquema que explica (**escribano2002introduccion**).

2.3. Fase I: Planificación

Es la fase en la que se define el alcance general del proyecto. En esta el cliente define lo que necesita mediante la redacción de HU y establece la prioridad de cada una. Luego, los programadores estiman los tiempos de desarrollo en base a esta información. Las estimaciones realizadas en esta fase son primarias, debido a que estarán basadas en datos de muy alto nivel y podrían variar cuando se analicen en cada ite- ración. Además se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses (**escribano2002introduccion**; **joskowicz2008reglas**).

2.3.1. Historias de Usuarios

Las HU son la técnica que utiliza XP para especificar los requisitos de software, estas deben ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a las tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra HU. Las estimaciones de esfuerzo, asociado a la implementación de las historias, la establecen los programadores, utilizando como medida el punto. Un punto, equivale a una semana ideal de programación (5 días laborables). Las historias, generalmente, valen de 1 a 3 puntos. (**escribano2002introduccion**). A continuación se describen las HU definidas parra llevar a cabo el desarrollo de los módulos.

Tabla 2.1. Historia de usuario # 1

Historia de usuario	
Número: 1	**Nombre:** Registro de Usuario
Usuario: Administrador, Usuario	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Bajo
Puntos estimados: 2	**Iteración asignada:** 1
Programador responsable: Técn. Carlos Brayan Rámila Chorens	

Continúa en la próxima página

Tabla 2.1. Continuación de la página anterior

Descripción: Como usuario nuevo, quiero poder registrarme en la aplicación proporcionando mi información personal, como nombre, dirección de correo electrónico y contraseña, para acceder a las funcionalidades de la aplicación. • nombre • dirección de correo electrónico • contraseña • carnet de identidad
Observaciones: • La aplicación debe permitir al usuario completar un formulario de registro con campos para nombre, correo electrónico y contraseña. • Los campos del formulario deben validar que la información ingresada sea válida y esté en el formato correcto. • Después de enviar el formulario, la aplicación debe crear una cuenta de usuario y almacenarla en la base de datos. • Se debe enviar un correo electrónico de confirmación al usuario registrado para verificar su dirección de correo electrónico. • El usuario debe poder iniciar sesión después de completar el registro exitosamente.

Tabla 2.2. Historia de usuario # 2

Historia de usuario	
Número: 2	**Nombre:** Iniciar Sesión
Usuario: Administrador, Usuario	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Bajo
Puntos estimados: 2	**Iteración asignada:** 1
Programador responsable: Técn. Carlos Brayan Rámila Chorens	
Descripción: Como usuario registrado, quiero poder iniciar sesión en la aplicación utilizando mi dirección de correo electrónico y contraseña, para acceder a mis datos y realizar acciones dentro de la aplicación. • dirección de correo electrónico • contraseña	

Continúa en la próxima página

Tabla 2.2. Continuación de la página anterior

Observaciones:

- La aplicación debe proporcionar un formulario de inicio de sesión con campos para correo electrónico y contraseña.
- Los campos del formulario deben validar que la información ingresada sea válida y coincida con los datos almacenados en la base de datos.
- Después de enviar el formulario, la aplicación debe autenticar al usuario y redirigirlo a la página principal si las credenciales son válidas.
- Si las credenciales son inválidas, la aplicación debe mostrar un mensaje de error al usuario indicando que las credenciales son incorrectas.
- La sesión del usuario debe mantenerse activa mientras navega por la aplicación, permitiéndole acceder a las funcionalidades protegidas sin necesidad de volver a iniciar sesión.

Tabla 2.3. Historia de usuario # 3

Historia de usuario	
Número: 3	**Nombre:** Gestión de Perfiles de Usuario
Usuario: Administrador	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Bajo
Puntos estimados: 1	**Iteración asignada:** 1
Programador responsable: Técn. Carlos Brayan Rámila Chorens	
Descripción: Como administrador, quiero poder gestionar los perfiles de usuario, incluyendo la capacidad de crear, editar y eliminar cuentas de usuario, para mantener el control sobre quién tiene acceso a la aplicación. • nombre • dirección de correo electrónico • contraseña • carnet de identidad	

Continúa en la próxima página

Tabla 2.3. Continuación de la página anterior

Observaciones:

- La aplicación debe proporcionar una interfaz de administración donde el administrador pueda ver una lista de todos los usuarios registrados.
- El administrador debe poder crear nuevos perfiles de usuario, proporcionando información como nombre, dirección de correo electrónico y contraseña.
- Se debe implementar la funcionalidad de edición para permitir al administrador actualizar la información de los perfiles de usuario existentes.
- El administrador debe poder eliminar cuentas de usuario cuando sea necesario, lo que resultará en la eliminación permanente de los datos asociados con esa cuenta.
- Se deben implementar controles de acceso adecuados para garantizar que solo el administrador tenga acceso a estas funcionalidades de gestión de perfiles.

2.3.2. Estimación de esfuerzo por Historia de Usuario

Se realiza la estimación de esfuerzo que arroja cada HU, con el objetivo de obtener un correcto desarrollo del sistema. Para una mayor organización se decide además, asignar a cada iteración el conjunto de historias agrupadas en correspondencia con el módulo al que representen. A continuación se muestra la estimación realizada:

2.3.3. Desarrollo del plan de iteraciones

Una vez definidas las HU y realizada una previa estimación de esfuerzos, se procede a la planificación de la etapa de implementación del sistema. En este espacio, se crea el plan de iteraciones, donde se especifica la prioridad con que se implementarán las HU organizadas por iteraciones. Teniendo en cuenta el esfuerzo asociado a las HU y a las prioridades del cliente, se define una versión que sea de valor para este.

Tabla 2.4. Estimación de esfuerzo por historia de usuario

Iteración	Historias de usuario		Puntos estimados (semanas)
1	1	Cargar Fichero	0.8
	2	Determinar Secuencia de Ensamble	0.2
	3	Gestionar restricción cambios de herramienta	0.5
	4	Esta es la prueba para el tinguiri	0.5
2	5	Graficar secuencias de ensamble	0.3
	6	Graficar secuencias de ensamble	0.3
	7	Graficar secuencias de ensamble	0.2
	8	Graficar secuencias de ensamble	0.2
3	9	Determinar Secuencia de Ensamble	2.5
4	10	Graficar secuencias de ensamble	1.1

Continúa en la próxima página

Tabla 2.4. Continuación de la página anterior

Total		**6.6**

2.3.4. Plan de duración de las iteraciones

A continuación, se presenta el plan de duración de las iteraciones. Este plan, tiene como finalidad, mostrar la duración de cada iteración, así como el orden en que serán implementadas las HU en cada iteración como se muestra en la tabla siguiente:

Tabla 2.5. Plan de duración de las iteraciones

Iteración	Historias de usuario		Duración (semanas)
1	1	Cargar Fichero	2.0
	2	Determinar Secuencia de Ensamble	
	3	Gestionar restricción cambios de herramienta	
	4	Esta es la prueba para el tinguiri	
2	5	Graficar secuencias de ensamble	1.0
	6	Graficar secuencias de ensamble	
	7	Graficar secuencias de ensamble	
	8	Graficar secuencias de ensamble	
3	9	Determinar Secuencia de Ensamble	2.5
4	10	Graficar secuencias de ensamble	1.1
Total			**6.6**

2.3.5. Plan de entregas

En el plan de entrega que se plantea a continuación, se hace una propuesta de las versiones (releases) del sistema. Cada versión se conformará al finalizar una iteración.

2.4. Fase II: Diseño del sistema

La plataforma se implementó siguiendo los principios del patrón arquitectónico MVT La arquitectura Modelo-Vista-Template (MVT) es un patrón de arquitectura de software utilizado por Django. Es una variante del patrón Modelo-Vista-Controlador (MVC) y se caracteriza por tratar de desacoplar lo máximo posible la interfaz de usuario de la lógica de la aplicación. El patrón MVT se divide en tres componentes principales:

- Modelo: El modelo es responsable de manejar los datos y la lógica de negocio de la aplicación. En otras palabras, el modelo es el encargado de interactuar con la base de datos y proporcionar los datos necesarios a la vista.

- Vista: La vista es responsable de presentar los datos al usuario final. En otras palabras, la vista es el encargado de manejar las interfaces gráficas y proporcionar información dinámica al usuario.
- Plantilla: La plantilla es responsable de definir cómo se presentan los datos en la vista. En otras palabras, el template define cómo se estructura y se muestra la información en la GUI.

El patrón MVT es una arquitectura de diseño que se utiliza para crear aplicaciones web eficientes y escalables. La separación de responsabilidades en los componentes del patrón permite que cada uno cumpla una función clara y definida, lo que facilita su mantenimiento y escalabilidad en el futuro. La reutilización de código, ya que, al separar las diferentes responsabilidades en componentes individuales, el código se puede escribir una vez y reutilizar en diferentes partes de la aplicación. Esto reduce el tiempo y el costo del desarrollo, y hace que el proceso de programación sea más eficiente.

2.5. Patrones de diseño

Los patrones de diseño, tratan los problemas que se repiten y que se presentan en situaciones particulares del diseño, con el fin de proponer soluciones a ellas. Se encargan de identificar clases, instancias, roles, colaboraciones entres estas, así como la distribución de responsabilidades. En resumen, es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular.

2.5.1. Patrones GRASP

GRASP son un conjunto de patrones de diseño de software orientado a objetos que se enfocan en asignar responsabilidades de manera adecuada a las clases y objetos en un sistema. Los patrones de diseño GRASP proporcionan un enfoque práctico para el diseño orientado a objetos y se pueden aplicar a diferentes etapas del ciclo de vida del software, desde la planificación hasta la implementación. Estos patrones de diseño pueden ayudar a mejorar el modularidad, la reutilización, la flexibilidad y la mantenibilidad del software, al tiempo que reducen la complejidad y el acoplamiento entre las clases.

Patrones de diseño GRASP:

- Controlador: maneja las solicitudes entrantes y enviar las respuestas correspondientes a los clientes. El controlador en Django se evidencia a través de una vista y en la configuración de las rutas URL, permitiendo una asignación efectiva de responsabilidades en la gestión de solicitudes y lógica de la aplicación.
- Creador: Django utiliza Creator para crear objetos de modelo. ORM de Django se encarga de crear objetos de modelo y mapearlos a la base de datos. La creación de instancias es unas de las actividades más comunes en un sistema orientado a objetos. El patrón se hace evidente en las clases contenidas dentro del archivo View.py, las cuales tienen la responsabilidad de instanciar los objetos creados con cada sistema de configuración básica.

- Experto: Django utiliza Expert para asignar responsabilidades de manera adecuada a las clases. En Django, las vistas son responsables de manejar las solicitudes entrantes y las plantillas son responsables de generar las respuestas HTML. Este patron se encuentra aplicado en todas las clases del archivo Models.py.
- Alta Cohesión: cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto identificable. Este patrón está representado en las clases contenidas dentro del archivo View.py, las cuales tienen una única responsabilidad que puede ser CreateView, UpdateView o DeleteView.

2.5.2. Patrones GoF

Los patrones de diseño GoF) son un conjunto de patrones de diseño de software orientado a objetos. Estos patrones de diseño se han convertido en un conjunto clásico de patrones de diseño y son ampliamente utilizados en el diseño y desarrollo de sistemas de software orientados a objetos. Proporcionan soluciones probadas y efectivas para problemas comunes de diseño de software orientado a objetos. Cada patrón describe un problema de diseño específico y proporciona una solución general que se puede aplicar a diferentes situaciones.

Patrones de diseño GoF:

- Decorador: agrega funcionalidades adicionales a las vistas. Django utiliza este patrón para la creación de vistas personalizadas y middleware, permitiendo agregar funcionalidad adicional a las vistas o el procesamiento de solicitudes sin modificar su código principal como la autenticación, caché, compresión.

Una vez definidos los patrones de diseño, se describen las clases utilizadas en la solución. Siguiendo la metodología XP, se analizan las HU y se descomponen en tareas independientes.

2.6. Tarjetas CRC

A continuación, las HU son evaluadas para dividirlas en tareas, cada tarea representa una característica distinta del sistema y se puede diseñar una prueba de unidad que verifique cada tarea, estas tareas se representan por medio de las tarjetas CRC.

Como se puede observar en la Figura 2.2, cada tarjeta contiene el nombre de la clase, una descripción de las responsabilidades o métodos asociados con la clase, así como la lista de las clases con que se relaciona o que colaboran con ella. A continuación se describen las tarjetas definidas para la implementación de la solución.

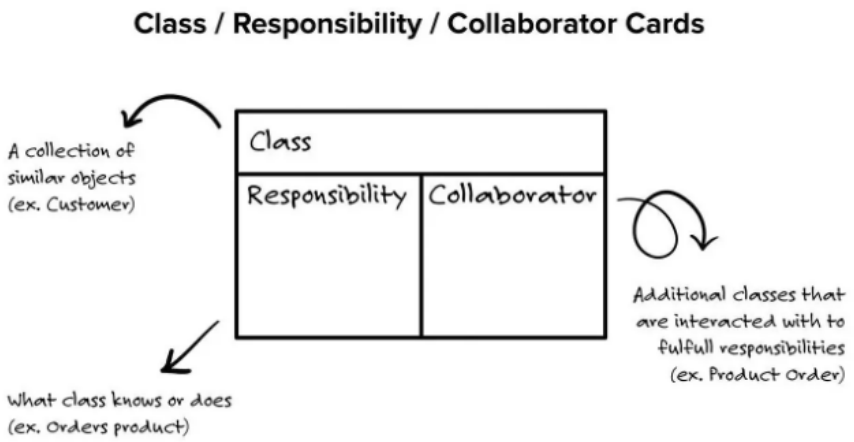

Figura 2.1. Prototipo de Tarjeta CRC

Tabla 2.6. Tarjeta CRC # 1

Tarjeta CRC	
Clase: RegistroUsuario	
Responsabilidad	**Colaboración**
• Recibir y validar la información del usuario para el registro. • Almacenar la información del usuario en la base de datos. • Enviar un correo electrónico de confirmación al usuario registrado.	• BaseDeDatos: para almacenar la información del usuario. • ServicioDeCorreo: para enviar el correo electrónico de confirmación.

Tabla 2.7. Tarjeta CRC # 2

Tarjeta CRC	
Clase: IniciarSesion	
Responsabilidad	**Colaboración**
• Autenticar al usuario utilizando su correo electrónico y contraseña. • Mantener la sesión activa mientras el usuario navega por la aplicación. • Redirigir al usuario a la página principal después de iniciar sesión correctamente.	• BaseDeDatos: para verificar las credenciales del usuario. • InterfazDeUsuario: para mostrar mensajes de error en caso de credenciales incorrectas.

Tabla 2.8. Tarjeta CRC # 3

Tarjeta CRC	
Clase: GestionPerfilesUsuario	
Responsabilidad	**Colaboración**
• Mostrar una lista de todos los usuarios registrados. • Permitir al administrador crear, editar y eliminar perfiles de usuario. • Implementar controles de acceso para garantizar la seguridad de las operaciones de gestión de perfiles.	• BaseDeDatos: para acceder y actualizar la información de los perfiles de usuario. • InterfazDeAdministrador: para mostrar la lista de usuarios y proporcionar opciones de edición y eliminación.

2.7. Conclusiones del capítulo

En este capítulo se han abordado los aspectos referentes a la concepción del producto a desarrollar y sus características funcionales. Esto permitió arribar a las siguientes conclusiones:

CAPÍTULO 3

Implementación y prueba del sistema

3.1. Introducción al capítulo

En el presente capítulo se detallan las iteraciones realizadas durante la etapa de construcción de los módulos propuestos, además se exponen las tareas de ingeniería generadas para cada HU que que fueron definidas, así como las pruebas de aceptación planificadas para el sistema. De esta forma es obtenido un producto funcional probado y listo para entregar al cliente al final de cada iteración como propone XP.

3.2. Fase III: Desarrollo

En esta fase, XP plantea que las HU seleccionadas para ser implementadas se realizan durante el transcurso de la iteración a la que pertenecen. Por estas razones, se lleva a cabo una revisión del plan de iteraciones y se modifican en caso de ser necesario. Como parte de este plan se descomponen las HU en tareas de ingeniería.

3.3. Tareas de ingeniería

Las tareas de ingeniería pueden estar descritas por un lenguaje técnico y no ser necesariamente entendibles por el cliente. Tienen como objetivo definir cada una de las actividades que dan cumplimiento a las HU, de forma tal que se entienda lo que el sistema tiene que hacer y facilite su construcción. Se describen algunas de las tareas de ingeniería correspondientes a las HU del sistema, el resto pueden ser consultadas en los anexos. Para una mayor organización, se definen en correspondencia con las iteraciones definidas como se manifiesta a continuación

3.3.1. Tareas de ingeniería para la Iteración I

3.3.2. Tareas de ingeniería para la Iteración II

3.3.3. Tareas de ingeniería para la Iteración III

Con las tareas de ingeniería definidas, se hace necesario establecer un conjunto de pruebas para comprobar la calidad de la solución implementada. Luego, se analizan estos casos de prueba y se ejecutan, lo que permite medir el nivel de cumplimiento con los objetivos de implementación trazados y el nivel de satisfacción del cliente. Así lo propone XP.

3.4. Fase IV: Pruebas

Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Por esta razón, se deben definir en el proceso de la ingeniería del software. Todo esto, contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones.
La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente.
Se decide realizar las pruebas de aceptación a los módulos implementados, debido a que el objetivo de estas, es verificar que el sistema cumpla con los requisitos establecidos por el usuario. De esta forma se puede obtener el grado de satisfacción del cliente.

3.5. Pruebas de aceptación

Las pruebas de aceptación son especificadas por el cliente, se centran en las características y funcionalidades generales del sistema que son visibles. Estas pruebas derivan de las HU que se han implementado como parte de la liberación del software. Una prueba de aceptación es como una caja negra. Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección y toma de decisiones acerca de estas pruebas. A continuación, se muestra una representación de las pruebas de aceptación a realizarse en cada iteración.

3.5.1. Pruebas de aceptación para la Iteración I

3.5.2. Pruebas de aceptación para la Iteración II

3.5.3. Pruebas de aceptación para la Iteración III

3.5.4. Análisis de las pruebas de aceptación

3.6. Conclusiones del capítulo

En este capítulo se especificó el proceso de implementación del sistema a partir del desglose de las HU en tareas de ingeniería, lo que permitió especificar los procedimientos necesarios para dar cumplimiento a cada HU. Además se definieron y aplicaron las pruebas de aceptación a las funcionalidades de los módulos desarrollados. Estas pruebas permitieron detectar una falla en el sistema y corregirla, lo que posibilitó mejorar la operabilidad del mismo.

Conclusiones

Escribir aquí las conclusiones.

Recomendaciones

Escribe aquí las recomendaciones de tu trabajo.

Generado con LaTeX: 20 de febrero de 2024: 12:33am

Apéndices