



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

Autores: Téc. Carlos Brayan Rámila Chorens

Alejandro Santana Viamontes

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

« Tu tiempo es limitado, de modo que no lo malgastes viviendo la vida de alguien distinto. No quedes atrapado en el dogma, que es vivir como otros piensan que deberías vivir. No dejes que los ruidos de las opiniones de los demás acallen tu propia voz interior. Y, lo que es más importante, ten el coraje para hacer lo que te dice tu corazón y tu intuición »

Steve Jobs

« Siempre me siento feliz. ¿Sabes por qué? Porque no espero nada de nadie; esperar siempre duele. Los problemas no son eternos, siempre tienen solución. Lo único que no se resuelve es la muerte. La vida es corta, por eso ámala, se feliz y siempre sonríe, solo vive intensamente. Antes de hablar, escucha. Antes de escribir, piensa. Antes de herir, siente. Antes de rendirte, intenta. Antes de morir, vive »

William Shakespeare

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, MSc. 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
Autor

Alejandro Santana Viamontes
Autor

MSc. Yadira Ramírez Rodríguez
Tutora

La implementación de la aplicación web para la mejora en la gestión de servicios de Transconsul [Sociedad Anónima \(S.A.\)](#) ha demostrado ser una solución efectiva para mejorar el proceso de gestión sus servicios. Para recopilar la información necesaria, se emplearon diferentes métodos teóricos y empíricos, y para llevar a cabo el desarrollo de la propuesta, fue seleccionado como marco de trabajo el framework Django de Python, y como Sistema Gestor de Base de Datos (SGBD), MySQL. Para guiar el desarrollo desde el análisis hasta las pruebas, se siguen los pasos que propone la metodología de desarrollo ágil Extreme Programming (XP). Se realizaron pruebas para validar la calidad y el cumplimiento de los requisitos. Esta solución tiene el potencial de reducir el tiempo , mejorando así su experiencia e imagen de los servicios de la empresa.

Palabras clave: gestión, servicio, cita, legal, documeto, cunsulta, asesoría, jurídica.

Introducción	1
1 Fundamentación Teórica	4
1.1 Introducción al capítulo	4
1.2 Sistema de gestión de citas y servicios	4
1.3 Sistemas Homólogos	5
1.4 Metodología de desarrollo de software	6
1.5 Herramientas y tecnología de desarrollo de software	7
2 Características y diseño del sistema	10
2.1 Introducción al capítulo	10
2.2 Propuesta de solución	11
2.3 Fase I: Planificación	12
2.3.1 Historias de Usuarios	12
2.3.2 Estimación de esfuerzo por Historia de Usuario	26
2.3.3 Desarrollo del plan de iteraciones	27
2.3.4 Plan de duración de las iteraciones	28
2.3.5 Plan de entregas	29
2.4 Fase II: Diseño del sistema	29
2.5 Patrones de diseño	30
2.5.1 Patrones General Responsibility Assignment Software Patterns (GRASP)	30
2.5.2 Patrones Gang of Four (GoF)	31
2.6 Tarjetas Class Responsibility Collaboration (CRC)	31
2.7 Conclusiones del capítulo	35
3 Implementación y prueba del sistema	37
3.1 Introducción al capítulo	37
3.2 Fase III: Desarrollo	37
3.3 Tareas de ingeniería	38
3.3.1 Tareas de ingeniería para la Iteración I	38

3.3.2	Tareas de ingeniería para la Iteración II	41
3.3.3	Tareas de ingeniería para la Iteración III	41
3.3.4	Tareas de ingeniería para la Iteración IV	43
3.4	Fase IV: Pruebas	45
3.5	Pruebas de aceptación	46
3.5.1	Pruebas de aceptación para la Iteración I	46
3.5.2	Pruebas de aceptación para la Iteración II	46
3.5.3	Pruebas de aceptación para la Iteración III	46
3.5.4	Análisis de las pruebas de aceptación	46
3.6	Conclusiones del capítulo	46
Conclusiones		47
Recomendaciones		48
Referencias bibliográficas		49
Apéndices		51

Índice de figuras

1.1	Pilares de la Metodología XP	7
2.1	Prototipo de Tarjeta CRC	32

Índice de tablas

1.1	Roles del sistema	5
1.2	Comparación de aplicaciones homólogas	6
2.1	Historia de usuario # 1	12
2.2	Historia de usuario # 2	13
2.3	Historia de usuario # 3	13
2.4	Historia de usuario # 4	14
2.5	Historia de usuario # 5	15
2.6	Historia de usuario # 6	16
2.7	Historia de usuario # 7	16
2.8	Historia de usuario # 8	16
2.9	Historia de usuario # 9	16
2.10	Historia de usuario # 10	17
2.11	Historia de usuario # 11	17
2.12	Historia de usuario # 12	18
2.13	Historia de usuario # 13	18
2.14	Historia de usuario # 14	18
2.15	Historia de usuario # 15	19
2.16	Historia de usuario # 16	19
2.17	Historia de usuario # 17	20
2.18	Historia de usuario # 18	20
2.19	Historia de usuario # 19	20
2.20	Historia de usuario # 20	21
2.21	Historia de usuario # 21	21
2.22	Historia de usuario # 22	21
2.23	Historia de usuario # 23	22
2.24	Historia de usuario # 24	22
2.25	Historia de usuario # 25	22
2.26	Historia de usuario # 26	23
2.27	Historia de usuario # 27	23

2.28	Historia de usuario # 28	23
2.29	Historia de usuario # 29	24
2.30	Historia de usuario # 30	24
2.31	Historia de usuario # 31	24
2.32	Historia de usuario # 32	25
2.33	Historia de usuario # 33	25
2.34	Historia de usuario # 34	25
2.35	Historia de usuario # 35	26
2.36	Historia de usuario # 36	26
2.37	Historia de usuario # 37	26
2.38	Estimación de esfuerzo por historia de usuario	27
2.39	Plan de duración de las iteraciones	28
2.40	Plan de entrega de versiones	29
2.41	Tarjeta CRC # 1	32
2.42	Tarjeta CRC # 2	32
2.43	Tarjeta CRC # 3	33
2.44	Tarjeta CRC # 4	33
2.45	Tarjeta CRC # 5	33
2.46	Tarjeta CRC # 6	34
2.47	Tarjeta CRC # 7	34
2.48	Tarjeta CRC # 8	35
3.1	Tarea de ingeniería # 1	38
3.2	Tarea de ingeniería # 2	38
3.3	Tarea de ingeniería # 3	39
3.4	Tarea de ingeniería # 4	39
3.5	Tarea de ingeniería # 5	39
3.6	Tarea de ingeniería # 6	40
3.7	Tarea de ingeniería # 7	40
3.8	Tarea de ingeniería # 8	40
3.9	Tarea de ingeniería # 9	41
3.10	Tarea de ingeniería # 10	41
3.11	Tarea de ingeniería # 11	41
3.12	Tarea de ingeniería # 12	42
3.13	Tarea de ingeniería # 13	42
3.14	Tarea de ingeniería # 14	42
3.15	Tarea de ingeniería # 15	43
3.16	Tarea de ingeniería # 16	43

3.17 Tarea de ingeniería # 17	44
3.18 Tarea de ingeniería # 18	44
3.19 Tarea de ingeniería # 19	44
3.20 Tarea de ingeniería # 20	45
3.21 Tarea de ingeniería # 21	45

Lista de algoritmos

Lista de códigos fuentes

El desarrollo acelerado de las tecnologías de la información y las telecomunicaciones ha marcado un antes y un después en la manera en que las empresas gestionan sus procesos y servicios. Este progreso ha permitido la creación de soluciones informáticas que automatizan tareas, optimizan recursos y mejoran la eficiencia operativa. En este sentido, los sistemas de gestión de citas y servicios han surgido como herramientas clave para facilitar la interacción entre las organizaciones y sus clientes, garantizando procesos más ágiles y efectivos.

En la actualidad, La sociedad civil de servicios Transconsul 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 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.

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. 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.

Además, existe una falta de control estadístico sobre los trámites realizados, el número de clientes atendidos y los ingresos obtenidos, lo que dificulta una evaluación eficiente del rendimiento semanal y mensual de la organización.

Ante esta situación, surge el problema de 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.

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.

El presente trabajo de diploma, está compuesto por tres capítulos, que incluyen los procedimientos desarrollados en relación con el trabajo investigativo, así como la propuesta de solución y validación de la investigación.

- **Capítulo 1 - Fundamentación Teórica:** en esta primera etapa se aborda el estado del arte del tema que se investiga. Se destaca un estudio bibliográfico detallado sobre los principales conceptos asociados a los mecanismos de servicios de consultas y asesorías legales con reserva de citas en línea que encuentran en la red de redes. En el transcurso de esta etapa, mientras se realiza el análisis del objeto de investigación, se seleccionan las tecnologías de desarrollo acordes para incorporar estas prestaciones en la plataforma Medicando.
- **Capítulo 2 - Características y diseño del sistema:** se describe el procedimiento seguido en las etapas de planificación y diseño que propone la metodología XP. Se especifican las Historias de Usuarios (HU), el plan de iteraciones y el plan de entregas. También se describe la propuesta de solución y se mencionan los patrones de diseño utilizados en la implementación de los módulos.
- **Capítulo 3 - Implementación y prueba del sistema:** en esta etapa se definen las tareas de ingeniería correspondientes a cada HU. También se aplican las pruebas de aceptación que permiten comprobar el correcto desarrollo de las funcionalidades implementadas.

1.1. Introducción al capítulo

El éxito de una solución tecnológica radica en la comprensión del problema y la correcta selección de enfoques metodológicos y herramientas. Este capítulo establece las bases teóricas que sustentan el desarrollo de la aplicación web propuesta para Transconsul [S.A.](#), con el fin de mejorar la gestión de sus servicios legales.

Se comienza con una revisión de los sistemas de gestión de citas y servicios, destacando su relevancia en la optimización de procesos y mejora de la atención al cliente. Además, se analizan sistemas homólogos implementados en otros entornos, extrayendo lecciones que enriquecen la propuesta.

A continuación, se presenta la metodología Extreme Programming (XP), elegida por su flexibilidad y enfoque en la entrega continua de valor, permitiendo una adaptación ágil a las necesidades cambiantes de Transconsul [S.A.](#).

Finalmente, se describen las herramientas tecnológicas seleccionadas, como Django y MySQL, justificando su elección por su escalabilidad, seguridad y capacidad de desarrollo ágil, lo que asegura una solución eficiente. Este marco teórico provee la estructura necesaria para el diseño y desarrollo del sistema, garantizando que las decisiones técnicas se alineen con los objetivos del proyecto.

1.2. Sistema de gestión de citas y servicios

Tabla 1.1. Roles del sistema

Roles	Descripción
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.
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

CONABI Consultores y Abogados Internacionales: Es un bufete cubano con más de 15 años de experiencia que ofrece servicios legales especializados en diversas áreas del Derecho, principalmente a clientes extranjeros o cubanos que residen fuera del país. Entre sus servicios más destacados está la asistencia en la obtención y legalización de certificaciones de antecedentes penales, un proceso que ahora también se puede realizar en línea. Los usuarios pueden registrarse en su portal web y pagar a través de plataformas cubanas como Transfermóvil y EnZona. Este servicio facilita la gestión remota de trámites, lo cual es útil tanto para residentes como para cubanos en el extranjero.([CONABI 2024](#))

CLAIM [S.A.](#) (Consultores Legales y Agentes de Invenciones y Marcas): Es una consultoría especializada en propiedad intelectual con sede en La Habana, Cuba. Su enfoque principal es asesorar y representar tanto a personas naturales como jurídicas, cubanas y extranjeras, en temas relacionados con la adquisición, mantenimiento, protección y defensa de derechos de propiedad industrial, derechos de autor, y registros sanitarios.CLAIM [S.A.](#) ofrece un sistema de gestión web centrado en la protección de la propiedad intelectual. Aunque no se ha especificado en detalle un sistema de gestión específico de tipo SaaS, CLAIM [S.A.](#) utiliza

tecnología avanzada para agilizar los procesos de registro, búsqueda, renovación y protección de derechos de propiedad industrial. A través de su plataforma, los clientes pueden gestionar registros de marcas, patentes y derechos de autor de manera eficiente, asegurando la confidencialidad y seguridad de la información que manejan.([CLAIM S.A 2024](#))

Clio: Es una plataforma de gestión legal bastante conocida. Facilita la administración de las operaciones diarias de los bufetes de abogados mediante herramientas que permiten gestionar casos, realizar un seguimiento del tiempo, facturar y programar citas. El portal del cliente es una característica clave, ya que permite a los clientes interactuar directamente con el bufete, agendar citas y acceder a información relevante sobre sus casos. Esto puede mejorar la eficiencia operativa y la satisfacción del cliente.([Clio | The Industry's #1 Legal Software | Try it for free! 2024](#))

Tabla 1.2. Comparación de aplicaciones homólogas

Aplicaciones	CONABI	CLAIM	Clio
Código Libre	No	No	No
Elemento 2	Descripción detallada del segundo elemento en la columna 2.	Descripción adicional en la columna 3.	Información adicional en la columna 4.
Elemento 3	Descripción detallada del tercer elemento en la columna 2.	Descripción adicional en la columna 3.	Información adicional en la columna 4.
Elemento 4	Descripción detallada del cuarto elemento en la columna 2.	Descripción adicional en la columna 3.	Información adicional en la columna 4.
Elemento 5	Descripción detallada del quinto elemento en la columna 2.	Descripción adicional en la columna 3.	Información adicional en la columna 4.

1.4. Metodología de desarrollo de software

Se siguieron los pasos que define la metodología [Programación Extrema \(XP\)](#) para guiar el proceso de desarrollo de software, atendiendo al esquema que se proponen en (Escribano, 2002), 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 (Joskowicz, 2008).

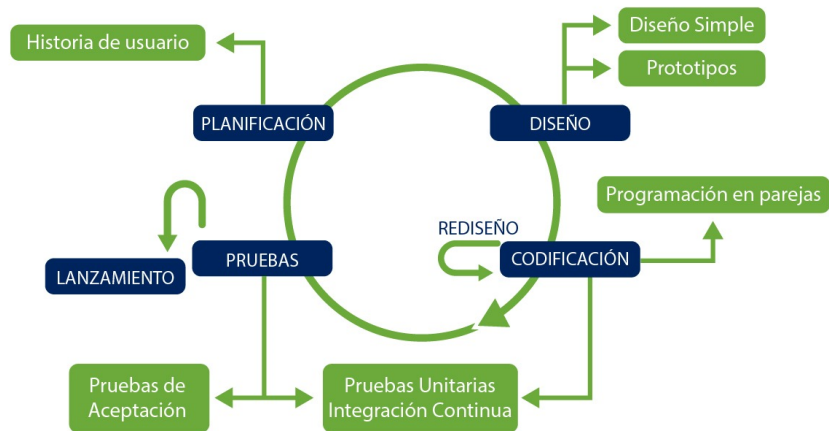


Figura 1.1. Pilares de la Metodología XP

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 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 click-jacking. 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 entornos 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 Martínez, [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, 2003).

Conclusiones del capítulo

En este capítulo se ha explorado el marco conceptual necesario para el desarrollo de un sistema de gestión de servicios legales en Transconsul S.A. . La comprensión de los conceptos clave y el uso adecuado de metodologías de desarrollo y herramientas tecnológicas son esenciales para optimizar la gestión de citas y servicios, mejorando la eficiencia y calidad de los procesos empresariales.

1. **La gestión eficiente de los servicios legales** es crucial para mantener la satisfacción del cliente y garantizar una operación fluida en empresas como Transconsul S.A.. La implementación de un sistema de gestión de citas permite reducir el tiempo de espera y facilita la interacción entre clientes y la organización.
2. **Los sistemas de gestión de servicios**, cuando se diseñan e implementan correctamente, no solo mejoran los procesos internos, sino que también contribuyen a elevar la reputación de la empresa, brindando una experiencia más ágil y accesible para los usuarios finales.
3. **Los sistemas homólogos** analizados en el capítulo proporcionaron valiosas lecciones sobre las características y funcionalidades clave que deben incluirse en un sistema de gestión de servicios para optimizar el rendimiento y la satisfacción del cliente.
4. **La adopción de la metodología Extreme Programming (XP)**, centrada en la entrega continua y el feedback del cliente, permite un enfoque flexible y ágil en el desarrollo del sistema. Además, el uso de herramientas como las *Historias de Usuario* y las tarjetas *CRC* facilitan la identificación de las necesidades y responsabilidades de cada módulo del sistema.

En resumen, este capítulo sienta las bases teóricas necesarias para garantizar que el sistema propuesto para Transconsul S.A. no solo sea técnicamente sólido, sino también alineado con las mejores prácticas de gestión de servicios y desarrollo ágil de software.

Características y diseño del sistema

2.1. Introducción al capítulo

En este capítulo se desarrolla la propuesta de solución tecnológica para enfrentar los problemas de gestión de servicios detectados en Transconsul [S.A.](#). A partir del análisis realizado en el capítulo anterior, se describe detalladamente el proceso de diseño del sistema, desde la definición de los módulos funcionales hasta la planificación de las entregas y las iteraciones de desarrollo.

La solución propuesta se fundamenta en la implementación de un sistema modular que permite gestionar de manera eficiente las citas y servicios ofrecidos por la empresa. En primer lugar, se identifican los roles clave que interactuarán con el sistema, desde administradores hasta usuarios finales, asegurando que cada actor del proceso cuente con las herramientas necesarias para realizar sus funciones de manera efectiva. Seguidamente, se introducen las Historias de Usuario (HU), que definen de manera clara los requisitos del sistema y las funcionalidades esperadas, proporcionando una visión centrada en el usuario y facilitando la priorización de las tareas de desarrollo.

Además, el capítulo abarca la planificación de las iteraciones, siguiendo los principios de la metodología ágil Extreme Programming (XP). Esto permite que el desarrollo del sistema sea flexible y adaptable a cambios en los requisitos o retroalimentación del cliente, garantizando la entrega de valor de manera continua. Se presenta un plan de iteraciones detallado que incluye la duración de cada fase, las HU asignadas y el esfuerzo estimado para cada tarea.

Finalmente, se aborda el diseño técnico del sistema, que incluye tanto la arquitectura general como los patrones de diseño utilizados. Se emplean patrones GRASP y GoF, los cuales garantizan que el sistema sea escalable, mantenible y eficiente. Este enfoque modular asegura que la solución pueda crecer y adaptarse a las necesidades cambiantes de la empresa sin comprometer su estabilidad o rendimiento.

Este capítulo ofrece una visión integral del diseño de la solución propuesta, asegurando que cada aspecto del desarrollo esté alineado con los objetivos de Transconsul S.A. y con las mejores prácticas en la creación de sistemas de gestión de servicios.

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 encargará 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 será el núcleo de la aplicación y se encargará de permitir a los usuarios programar, aceptar y cancelar citas teniendo en cuenta la disponibilidad diaria de las mismas. Deberá incluir notificaciones de citas programadas y recordatorios automáticos.
- **Gestión de Clientes:** Manejará toda la información relacionada con los clientes, incluyendo su perfil, historial de citas y otra información relevante.
- **Gestión de Reseñas:** Permite a los usuarios dejar reseñas sobre su experiencia con el servicio, las cuales podrán ser visualizadas y gestionadas por el administrador.
- **Gestión de Trabajadores:** Permite a los administradores gestionar la información de los trabajadores y una reseña sobre ellos.
- **Gestión de tramites:** Permite a los administradores crear, modificar y eliminar trámites disponibles para los usuarios.
- **Comunicación y Notificaciones:** Este módulo será responsable de facilitar la comunicación entre el bufete de abogados y sus clientes, a través de mensajes de correos electrónicos que informaran el estado de las citas y servicios solicitados.
- **Administración del Sistema:** Este módulo será 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 encargará 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 (Escribano, 2002).

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 iteración. 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 (Escribano, 2002; Joskowicz, 2008).

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. (Escribano, 2002). A continuación se describen las HU definidas para 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: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.2	Iteración asignada: 1
Programador responsable: Alejandro Santana Viamontes	
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. <ul style="list-style-type: none"> • nombre • dirección de correo electrónico • contraseña • rectificar contraseña 	

Continúa en la próxima página

Tabla 2.1. Continuación de la página anterior

Observaciones: <ul style="list-style-type: none"> • 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: Página de Inicio
Usuario: Administrador, Legalizador, Cliente, Notario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite a los usuarios acceder a la pagina de inicio del sistema desde donde se accede a las demás funcionalidades del sistema.	
Observaciones:	

Tabla 2.3. Historia de usuario # 3

Historia de usuario	
Número: 3	Nombre: Iniciar Sesión
Usuario: Administrador, Legalizador, Cliente, Notario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 1
Programador responsable: Alejandro Santana Viamontes	
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. <ul style="list-style-type: none"> • dirección de correo electrónico • contraseña 	

Continúa en la próxima página

Tabla 2.3. Continuación de la página anterior

<p>Observaciones:</p> <ul style="list-style-type: none"> • 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 2.4. Historia de usuario # 4

Historia de usuario	
Número: 4	Nombre: Gestión de Perfiles de Usuario
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 1
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
<p>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.</p> <ul style="list-style-type: none"> • nombre • dirección de correo electrónico • contraseña • carnet de identidad • rol 	

Continúa en la próxima página

Tabla 2.4. Continuación de la página anterior

<p>Observaciones:</p> <ul style="list-style-type: none"> • 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, así como el rol. • 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.

Tabla 2.5. Historia de usuario # 5

Historia de usuario	
Número: 5	Nombre: Reservar Cita
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
<p>Descripción: Permite al usuario acceder a la opción de registrar su cita, donde se muestra una página con un formulario a rellenar con los siguientes campos</p> <ul style="list-style-type: none"> • Cantidad de Documentos a Legalizar • Obtención de documentos • Fecha • Trámite • Con Notario • Motivo 	
<p>Observaciones:</p> <ul style="list-style-type: none"> • El usuario debe estar autenticado en el sistema y con la información de su perfil completa. • El sistema comprueba que haya disponibilidad para la fecha seleccionada y guarde al reserva en la base de datos. 	

Tabla 2.6. Historia de usuario # 6

Historia de usuario	
Número: 6	Nombre: Listar Citas
Usuario: Notario,Legalizador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 2
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite a los usuarios listar las citas de todos los clientes donde se muestran la fecha, el cliente, estado y los botones .^ceptarz "Denegar"	
Observaciones:	

Tabla 2.7. Historia de usuario # 7

Historia de usuario	
Número: 7	Nombre: Ver Citas
Usuario: Cliente	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0.3	Iteración asignada: 2
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al cliente listar sus citas donde se muestran la fecha y el estado.	
Observaciones:	

Tabla 2.8. Historia de usuario # 8

Historia de usuario	
Número: 8	Nombre: Aceptar Citas
Usuario: Notario,Legalizador	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0.1	Iteración asignada: 2
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario cambiar el estado de la cita a .^ceptada"lo que notificará al cliente a su correo electrónico.	
Observaciones:	

Tabla 2.9. Historia de usuario # 9

Historia de usuario	
Número: 9	Nombre: Denegar Citas
Usuario: Notario,Legalizador	
Prioridad en negocio: Media	Riesgo en desarrollo: Media

Continúa en la próxima página

Tabla 2.9. Continuación de la página anterior

Puntos estimados: 0.1	Iteración asignada: 2
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario cambiar el estado de la cita a "Denegada" lo que notificará al cliente a su correo electrónico.	
Observaciones:	

Tabla 2.10. Historia de usuario # 10

Historia de usuario	
Número: 10	Nombre: Modificar Perfil Usuario
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Téc. Carlos Brayan Rámila Chorens, Alejandro Santana Viamontes	
Descripción: Permite al usuario poder acceder a la opción de modificar usuario, donde se muestra una página con un formulario con los datos que el usuario puede modificar los cuales son. <ul style="list-style-type: none"> • Nombre • Apellidos • Usuario • Dirección • Teléfono • Carnet de identidad • Foto de perfil 	
Observaciones:	

Tabla 2.11. Historia de usuario # 11

Historia de usuario	
Número: 11	Nombre: Eliminar Usuario
Usuario: Administrador	
Prioridad en negocio: Medio	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario poder acceder a la opción de eliminar uno o varios usuarios seleccionados, donde se muestra una advertencia con un mensaje de: ¿Esta seguro? No podrá revertir esta acción	
Observaciones:	

Tabla 2.12. Historia de usuario # 12

Historia de usuario	
Número: 12	Nombre: Crear Tramite
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario poder acceder a la opción de crear un trámite a realizar, donde se muestra una página con un formulario a rellenar con los siguientes datos: <ul style="list-style-type: none"> • Código • Nombre • Descripción • Precio 	
Observaciones:	

Tabla 2.13. Historia de usuario # 13

Historia de usuario	
Número: 13	Nombre: Modificar Tramite
Usuario: Administrador	
Prioridad en negocio: Bajo	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario poder acceder a la opción de modificar un trámite a realizar, donde se muestra una página con un formulario con los siguientes datos: <ul style="list-style-type: none"> • Código • Nombre • Descripción • Precio 	
Observaciones:	

Tabla 2.14. Historia de usuario # 14

Historia de usuario	
Número: 14	Nombre: Eliminar Tramite
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio

Continúa en la próxima página

Tabla 2.14. Continuación de la página anterior

Puntos estimados: 0.3	Iteración asignada: 2
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario poder acceder a la opción de eliminar trámite o los seleccionados, donde se muestra una advertencia con un mensaje: ¿Esta seguro? No podrá revertir esta acción	
Observaciones:	

Tabla 2.15. Historia de usuario # 15

Historia de usuario	
Número: 15	Nombre: Listar Trámites
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 2
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario listar los trámites, donde se muestra, el código, nombre, descripción, precio y los botones de "Agregar", "Modificar" y "Eliminar"	
Observaciones:	

Tabla 2.16. Historia de usuario # 16

Historia de usuario	
Número: 16	Nombre: Crear Trabajador
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Alejandro Santana Viamontes	
Descripción: Permite al usuario poder acceder a la opción de crear trabajador, donde se muestra una página con un formulario a rellenar con los siguientes datos: <ul style="list-style-type: none"> • Nombre • Nivel • Descripción • Foto 	
Observaciones:	

Tabla 2.17. Historia de usuario # 17

Historia de usuario	
Número: 17	Nombre: Modificar Trabajador
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Alejandro Santana Viamontes	
Descripción: Permite al usuario poder acceder a la opción de modificar trabajador, donde se muestra una página con un formulario a rellenar con los siguientes datos: <ul style="list-style-type: none"> • Nombre • Nivel • Descripción • Foto 	
Observaciones:	

Tabla 2.18. Historia de usuario # 18

Historia de usuario	
Número: 18	Nombre: Eliminar Trabajador
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Alejandro Santana Viamontes	
Descripción: Permite al usuario poder acceder a la opción de eliminar trabajador, donde se muestra una página con un mensaje de advertencia: ¿Esta seguro? No podrá revertir esta acción	
Observaciones:	

Tabla 2.19. Historia de usuario # 19

Historia de usuario	
Número: 19	Nombre: Listar Trabajadores
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Alejandro Santana Viamontes	
Descripción: Permite al usuario listar los trabajadores, donde se muestra el nombre, nivel y descripción además de los botones de "Agregar", "Modificar" y "Eliminar"	
Observaciones:	

Tabla 2.20. Historia de usuario # 20

Historia de usuario	
Número: 20	Nombre: Ver Trabajador
Usuario: Cliente	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Alejandro Santana Viamontes	
Descripción: Permite al usuario ver la información de los trabajadores a través de la pagina de inicio.	
Observaciones:	

Tabla 2.21. Historia de usuario # 21

Historia de usuario	
Número: 21	Nombre: Crear Artículo
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario poder crear artículos para el Blog completando un formulario con los siguientes campos. <ul style="list-style-type: none"> • Título • Foto • Descripción 	
Observaciones:	

Tabla 2.22. Historia de usuario # 22

Historia de usuario	
Número: 22	Nombre: Modificar Artículo
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Téc. Carlos Brayan Rámila Chorens	

Continúa en la próxima página

Tabla 2.22. Continuación de la página anterior

Descripción: Permite al usuario poder modificar los artículos del Blog completando un formulario con los siguientes campos. <ul style="list-style-type: none"> • Título • Foto • Descripción
Observaciones:

Tabla 2.23. Historia de usuario # 23

Historia de usuario	
Número: 23	Nombre: Eliminar Artículo
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario poder acceder a la opción de eliminar artículo, donde se muestra una página con un mensaje de advertencia: ¿Esta seguro? No podrá revertir esta acción	
Observaciones:	

Tabla 2.24. Historia de usuario # 24

Historia de usuario	
Número: 24	Nombre: Listar Artículos
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario listar los artículos, donde se muestra el nombre y la fecha de creación además de los botones de Crear", Editar", Eliminar"	
Observaciones:	

Tabla 2.25. Historia de usuario # 25

Historia de usuario	
Número: 25	Nombre: Leer Artículo
Usuario: Cliente	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 3

Continúa en la próxima página

Tabla 2.25. Continuación de la página anterior

Programador responsable: Téc. Carlos Brayan Rámila Chorens
Descripción: Permite al usuario leer los artículos del blog.
Observaciones:

Tabla 2.26. Historia de usuario # 26

Historia de usuario	
Número: 26	Nombre: Previsualizar Artículo
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario observar como quedará el artículo una vez publicado.	
Observaciones:	

Tabla 2.27. Historia de usuario # 27

Historia de usuario	
Número: 27	Nombre: Crear Disponibilidad
Usuario: Notario, Legalizador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario crear una disponibilidad completando un formulario con los siguientes campos. <ul style="list-style-type: none"> • Fecha • Cantidad 	
Observaciones:	

Tabla 2.28. Historia de usuario # 28

Historia de usuario	
Número: 28	Nombre: Modificar Disponibilidad
Usuario: Notario, Legalizador	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Téc. Carlos Brayan Rámila Chorens	

Continúa en la próxima página

Tabla 2.28. Continuación de la página anterior

Descripción: Permite al usuario modificar una disponibilidad completando un formulario con los siguientes campos. <ul style="list-style-type: none"> • Fecha • Cantidad
Observaciones:

Tabla 2.29. Historia de usuario # 29

Historia de usuario	
Número: 29	Nombre: Eliminar Disponibilidad
Usuario: Notario,Legalizador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario acceder a la opción de eliminar disponibilidad, donde se muestra una página con un mensaje de advertencia: ¿Esta seguro? No podrá revertir esta acción	
Observaciones:	

Tabla 2.30. Historia de usuario # 30

Historia de usuario	
Número: 30	Nombre: Listar Disponibilidades
Usuario: Notario,Legalizador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario listar las disponibilidades, donde se muestra la fecha, cantidad disponibles y cantidad reservadas, además de los botones de "Crear", "Editar" y "Eliminar"	
Observaciones:	

Tabla 2.31. Historia de usuario # 31

Historia de usuario	
Número: 31	Nombre: Crear Opinión
Usuario: Clientes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Téc. Carlos Brayan Rámila Chorens	

Continúa en la próxima página

Tabla 2.31. Continuación de la página anterior

Descripción: Permite al usuario crear una opinión sobre los servicios brindados por la empresa.
Observaciones:

Tabla 2.32. Historia de usuario # 32

Historia de usuario	
Número: 32	Nombre: Eliminar Opinión
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario acceder a la opción de eliminar opinión, donde se muestra una página con un mensaje de advertencia: ¿Esta seguro? No podrá revertir esta acción	
Observaciones:	

Tabla 2.33. Historia de usuario # 33

Historia de usuario	
Número: 33	Nombre: Listar Opiniones
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario listar las opiniones hechas por los clientes, donde se muestra la información de las opiniones así como el botón "Eliminar"	
Observaciones:	

Tabla 2.34. Historia de usuario # 34

Historia de usuario	
Número: 34	Nombre: Acceder al blog
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario acceder al blog donde podrá ver los artículos publicados.	
Observaciones:	

Tabla 2.35. Historia de usuario # 35

Historia de usuario	
Número: 35	Nombre: Acceder a Contacto
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario acceder a la información de contacto como dirección o teléfono de la empresa, un mapa con la ubicación y el formulario de soporte.	
Observaciones:	

Tabla 2.36. Historia de usuario # 36

Historia de usuario	
Número: 36	Nombre: Acceder a Servicios
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario acceder a los servicios que brinda la empresa donde se muestran los servicios una breve descripción de estos.	
Observaciones:	

Tabla 2.37. Historia de usuario # 37

Historia de usuario	
Número: 37	Nombre: Acceder a Nosotros
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.2	Iteración asignada: 4
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Permite al usuario acceder a una reseña de la empresa donde se muestran datos como los años de experiencia, la misión y la visión de la empresa, entre otros.	
Observaciones:	

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 2.38. Estimación de esfuerzo por historia de usuario

Iteración	Historias de usuario		Puntos estimados (semanas)
1	1	Registro de Usuario	0.2
	2	Página de Inicio	1.0
	3	Iniciar Sesión	0.2
	4	Gestión de Perfiles de Usuario	0.2
	5	Reservar Cita	1.0
	6	Modificar Perfil Usuario	0.5
2	7	Listar Citas	0.3
	8	Ver Citas	0.3
	9	Aceptar Citas	0.1
	10	Denegar Citas	0.1
	11	Eliminar Usuario	0.5
	12	Crear Tramite	1.0
	13	Modificar Tramite	1.0
	14	Eliminar Tramite	0.3
	15	Listar Tramites	0.3
3	16	Crear Trabajador	0.2
	17	Modificar Trabajador	0.2
	18	Eliminar Trabajador	0.2
	19	Listar Trabajadores	0.2
	20	Ver Trabajador	0.2
	21	Crear Artículo	0.2
	22	Modificar Artículo	0.2
	23	Eliminar Artículo	0.2
	24	Listar Artículos	0.2
	25	Leer Artículo	0.2
	26	Previsualizar Artículo	0.2
	27	Crear Disponibilidad	0.2
	28	Modificar Disponibilidad	0.2
	29	Eliminar Disponibilidad	0.2
	30	Listar Disponibilidades	0.2

Continúa en la próxima página

Tabla 2.38. Continuación de la página anterior

31	Crear Opinión	0.2
32	Eliminar Opinión	0.2
33	Listar Opiniones	0.2
34	Acceder al blog	0.2
35	Acceder a Contacto	0.2
36	Acceder a Servicios	0.2
37	Acceder a Nosotros	0.2
Total		11.4

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 2.39. Plan de duración de las iteraciones

Iteración	Historias de usuario		Duración (semanas)
1	1	Registro de Usuario	3.1
	2	Página de Inicio	
	3	Iniciar Sesión	
	4	Gestión de Perfiles de Usuario	
	5	Reservar Cita	
	6	Modificar Perfil Usuario	
2	7	Listar Citas	3.9
	8	Ver Citas	
	9	Aceptar Citas	
	10	Denegar Citas	
	11	Eliminar Usuario	
	12	Crear Tramite	
	13	Modificar Tramite	
	14	Eliminar Tramite	
	15	Listar Tramites	
3	16	Crear Trabajador	2.2
	17	Modificar Trabajador	
	18	Eliminar Trabajador	
	19	Listar Trabajadores	
	20	Ver Trabajador	
	21	Crear Artículo	
	22	Modificar Artículo	

Continúa en la próxima página

Tabla 2.39. Continuación de la página anterior

	23	Eliminar Artículo	
	24	Listar Artículos	
	25	Leer Artículo	
	26	Previsualizar Artículo	
4	27	Crear Disponibilidad	2.2
	28	Modificar Disponibilidad	
	29	Eliminar Disponibilidad	
	30	Listar Disponibilidades	
	31	Crear Opinión	
	32	Eliminar Opinión	
	33	Listar Opiniones	
	34	Acceder al blog	
	35	Acceder a Contacto	
	36	Acceder a Servicios	
	37	Acceder a Nosotros	
Total			11.4

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.

Tabla 2.40. Plan de entrega de versiones

Entregable	Iteración 1	Iteración 2	Iteración 3	Iteración 4
Módulos	Versión 0.1	Versión 0.2	Versión 0.3	Versión 0.4
Fecha	20/9/2024	20/10/2024	5/11/2024	20/11/2024

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 (Sarmiento, 2019). 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 (Rafael D. Hernandez, 2021).
- 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 (Sarmiento, 2019).

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 entre 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 la modularidad, la reutilización, la flexibilidad y la mantenibilidad del software, al tiempo que reducen la complejidad y el acoplamiento entre las clases (CodeScouts, 2022).

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 una 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 patrón 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 (CodeScouts, [2022](#)).

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 (Gamma; Helm; Johnson y Vlissides, [1994](#)).

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 (Silva, [2019](#)).

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 dividir las 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.

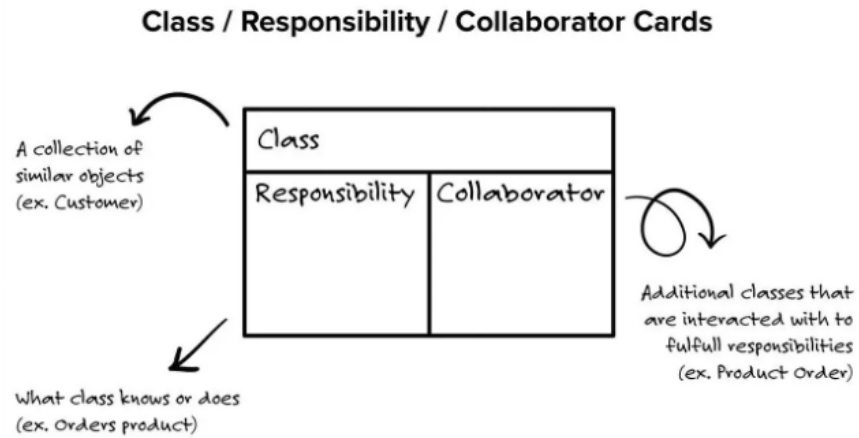


Figura 2.1. Prototipo de Tarjeta CRC

Tabla 2.41. Tarjeta CRC # 1

Tarjeta CRC	
Clase: workerView	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • worker() • storeWorker() • updateWorker() • deleteWorker() • deleteAllWorker() 	<ul style="list-style-type: none"> • Worker • User • Middleware

Tabla 2.42. Tarjeta CRC # 2

Tarjeta CRC	
Clase: opinionView	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • opinion() • storeOpinion() • deleteOpinion() • deleteAllOpinion() 	<ul style="list-style-type: none"> • Opinion • User • Middleware

Tabla 2.43. Tarjeta CRC # 3

Tarjeta CRC	
Clase: userView	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • user() • storeUser() • updateUser() • deleteUser() • deleteAllUser() 	<ul style="list-style-type: none"> • User • Middleware

Tabla 2.44. Tarjeta CRC # 4

Tarjeta CRC	
Clase: webView	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • home() • service() • about() • testimoniof() • blog() • dashboard() 	<ul style="list-style-type: none"> • User • Procedure • Opinion • Worker • Middleware

Tabla 2.45. Tarjeta CRC # 5

Tarjeta CRC	
Clase: profileView	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • profile() • signup() • signin() • signout() • changePassword() • updatePerfil() • updatePhotoPerfil() 	<ul style="list-style-type: none"> • User • Middleware

Tabla 2.46. Tarjeta CRC # 6

Tarjeta CRC	
Clase: dateView	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • reserve() • date() • showDate() • storeDate() • aceptarDate() • cancelarDate() 	<ul style="list-style-type: none"> • Date • User • Available • Procedure • Middleware

Tabla 2.47. Tarjeta CRC # 7

Tarjeta CRC	
Clase: procedureView	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • procedure() • storeProcedure() • updateProcedure() • deleteProcedure() • deleteAllProcedure() 	<ul style="list-style-type: none"> • Procedure • Middleware

Tabla 2.48. Tarjeta CRC # 8

Tarjeta CRC	
Clase: availableView	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • available() • storeAvailable() • updateAvailable() • deleteAvailable() • deleteAllAvailable() 	<ul style="list-style-type: none"> • Available • Middleware

2.7. Conclusiones del capítulo

En el presente capítulo se ha descrito en detalle el diseño y las características principales del sistema propuesto para Transconsul S.A., con el objetivo de mejorar la gestión de sus servicios legales. A través de la implementación de un enfoque modular, una metodología ágil y herramientas de diseño avanzadas,

se ha logrado establecer una solución técnica robusta y alineada con los requerimientos de la empresa. A continuación, se destacan las conclusiones principales:

1. **Diseño modular y flexible:** El sistema propuesto se basa en un diseño modular que permite una implementación escalonada de las funcionalidades, facilitando tanto el desarrollo como el mantenimiento. Esta estructura asegura que el sistema pueda adaptarse a las necesidades cambiantes de la empresa y escalar conforme se incrementen los servicios ofrecidos o se modifiquen los procesos.
2. **Historias de Usuario (HU) como guía central:** La definición clara y detallada de las Historias de Usuario (HU) ha sido fundamental para captar de manera precisa los requerimientos de los diferentes actores del sistema, lo que ha permitido priorizar las funcionalidades críticas y garantizar que el desarrollo esté centrado en las necesidades del cliente final.
3. **Planificación iterativa con XP:** La metodología ágil **Extreme Programming (XP)** ha proporcionado un marco de trabajo eficiente, que no solo permite un desarrollo continuo, sino también una retroalimentación constante con el cliente, asegurando que los cambios puedan incorporarse rápidamente sin afectar la calidad del producto final. El plan de iteraciones y el plan de entregas garantizan un desarrollo ordenado y progresivo.
4. **Uso de patrones de diseño (GRASP y GoF):** La aplicación de patrones de diseño, como **GRASP** y **GoF**, ha contribuido significativamente a una mejor organización del código y a una asignación más eficiente de las responsabilidades entre las clases del sistema. Estos patrones garantizan un diseño coherente, reutilizable y fácil de mantener, reduciendo la complejidad a medida que el sistema crece.

En conclusión, este capítulo ha establecido las bases técnicas y funcionales necesarias para la implementación del sistema, asegurando que la solución propuesta no solo cumpla con los objetivos establecidos, sino que también sea lo suficientemente flexible y escalable para evolucionar junto con las necesidades de Transconsul [S.A.](#).

Implementación y prueba del sistema

3.1. Introducción al capítulo

Este capítulo se centra en la implementación y validación del sistema de gestión de citas desarrollado para Transconsul S.A., siguiendo las pautas establecidas en los capítulos previos. La implementación se realizó utilizando la metodología ágil Extreme Programming (XP), lo que permitió un enfoque iterativo y adaptable, favoreciendo la retroalimentación continua por parte del cliente y asegurando que el sistema evolucionara conforme a sus necesidades.

El proceso de desarrollo estuvo guiado por las Historias de Usuario (HU), que sirvieron para desglosar las funcionalidades principales en tareas de ingeniería específicas. Estas tareas se abordaron en distintas iteraciones, lo que facilitó la construcción modular del sistema y permitió la entrega de incrementos funcionales en cada fase del proyecto. A lo largo del desarrollo, se prestó especial atención a garantizar que cada componente cumpliera con los requisitos funcionales, técnicos y de experiencia de usuario planteados inicialmente.

Una vez completada la implementación, el sistema fue sometido a un riguroso proceso de pruebas, que incluyó pruebas unitarias, pruebas de integración y pruebas de aceptación. Estas pruebas fueron fundamentales para asegurar el correcto funcionamiento del sistema en su conjunto y garantizar que cada módulo se integrara sin problemas, cumpliendo con los estándares de calidad y rendimiento esperados.

3.2. Fase III: Desarrollo

El desarrollo del sistema de gestión de citas para Transconsul S.A. se llevó a cabo siguiendo una metodología ágil, basada en Extreme Programming (XP), que permitió organizar el trabajo en iteraciones cortas y entregas parciales de funcionalidades. Esta metodología fue seleccionada por su capacidad para adaptarse a

cambios en los requisitos y su enfoque en la retroalimentación continua del cliente. El proceso de desarrollo se estructuró en varias fases, cada una de ellas enfocada en la implementación de módulos específicos.

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

Tabla 3.1. Tarea de ingeniería # 1

Tarea	
Número de tarea: 1	Número de Historia de usuario: 1
Nombre de la tarea: Implementar el formulario de registro de usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 1 de septiembre de 2024	Fecha de fin: 2 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se implementa una funcionalidad que permite a un nuevo usuario registrarse en la aplicación proporcionando su nombre, correo electrónico, contraseña y carnet de identidad. El sistema validará la información ingresada y almacenará los datos del usuario en la base de datos. Una vez completado el registro, el sistema enviará un correo electrónico de confirmación para verificar la dirección de correo del usuario y permitir que este pueda iniciar sesión en la aplicación.	

Tabla 3.2. Tarea de ingeniería # 2

Tarea	
Número de tarea: 2	Número de Historia de usuario: 1
Nombre de la tarea: Implementar validación de los campos de registro	
Tipo de tarea: Desarrollo	Puntos estimados: 1.2
Fecha de inicio: 2 de septiembre de 2024	Fecha de fin: 3 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	

Continúa en la próxima página

Tabla 3.2. Continuación de la página anterior

Descripción: Se implementan las validaciones tanto en el frontend como en el backend para los campos de registro. El correo electrónico debe tener el formato adecuado, la contraseña debe cumplir con los requisitos de seguridad, y el número de carnet de identidad debe ser válido según el formato establecido por las autoridades. El sistema notificará al usuario si alguna información es incorrecta, solicitando que la corrija antes de completar el registro.
--

Tabla 3.3. Tarea de ingeniería # 3

Tarea	
Número de tarea: 3	Número de Historia de usuario: 1
Nombre de la tarea: Almacenamiento de la cuenta de usuario en la base de datos	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha de inicio: 3 de septiembre de 2024	Fecha de fin: 4 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se desarrolla la funcionalidad para almacenar los datos del nuevo usuario en la base de datos, utilizando el modelo de datos que incluye nombre, correo electrónico, contraseña encriptada y carnet de identidad. Los datos se almacenan de forma segura, y el sistema garantiza que no haya duplicados en las direcciones de correo o los números de identificación.	

Tabla 3.4. Tarea de ingeniería # 4

Tarea	
Número de tarea: 4	Número de Historia de usuario: 2
Nombre de la tarea: Implementar el formulario de inicio de sesión	
Tipo de tarea: Desarrollo	Puntos estimados: 0.8
Fecha de inicio: 5 de septiembre de 2024	Fecha de fin: 6 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se implementa un formulario de inicio de sesión que permitirá a los usuarios ingresar su dirección de correo electrónico y contraseña. El formulario debe validar que ambos campos estén presentes y que el formato del correo electrónico sea correcto.	

Tabla 3.5. Tarea de ingeniería # 5

Tarea	
Número de tarea: 5	Número de Historia de usuario: 2
Nombre de la tarea: Validación de las credenciales de inicio de sesión	
Tipo de tarea: Desarrollo	Puntos estimados: 1.2
Fecha de inicio: 6 de septiembre de 2024	Fecha de fin: 7 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	

Continúa en la próxima página

Tabla 3.5. Continuación de la página anterior

Descripción: Se implementa la lógica de validación de credenciales, que verificará si el correo electrónico y la contraseña ingresados coinciden con los datos almacenados en la base de datos. Si las credenciales son válidas, el sistema debe autenticar al usuario y redirigirlo a la página principal. Si las credenciales son incorrectas, se mostrará un mensaje de error.
--

Tabla 3.6. Tarea de ingeniería # 6

Tarea	
Número de tarea: 6	Número de Historia de usuario: 2
Nombre de la tarea: Autenticación de sesión de usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 7 de septiembre de 2024	Fecha de fin: 8 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se implementa el mecanismo de autenticación para generar una sesión de usuario una vez que el inicio de sesión sea exitoso. La sesión debe mantenerse activa mientras el usuario navega por la aplicación, y solo debe finalizar cuando el usuario cierre sesión manualmente o la sesión expire por inactividad.	

Tabla 3.7. Tarea de ingeniería # 7

Tarea	
Número de tarea: 7	Número de Historia de usuario: 2
Nombre de la tarea: Mensajes de error para credenciales incorrectas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 8 de septiembre de 2024	Fecha de fin: 9 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se desarrolla la funcionalidad para mostrar un mensaje de error cuando las credenciales de inicio de sesión sean incorrectas. El mensaje debe ser claro y específico, indicando al usuario que su correo o contraseña no son válidos, y sugiriendo la recuperación de su contraseña.	

Tabla 3.8. Tarea de ingeniería # 8

Tarea	
Número de tarea: 8	Número de Historia de usuario: 2
Nombre de la tarea: Mensajes de error para credenciales incorrectas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 8 de septiembre de 2024	Fecha de fin: 9 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	

Continúa en la próxima página

Tabla 3.8. Continuación de la página anterior

Descripción: Se desarrolla la funcionalidad para mostrar un mensaje de error cuando las credenciales de inicio de sesión sean incorrectas. El mensaje debe ser claro y específico, indicando al usuario que su correo o contraseña no son válidos, y sugiriendo la recuperación de su contraseña.
--

3.3.2. Tareas de ingeniería para la Iteración II

Tabla 3.9. Tarea de ingeniería # 9

Tarea	
Número de tarea: 9	Número de Historia de usuario: 7
Nombre de la tarea: Implementar la integración con el servicio SMS	
Tipo de tarea: Desarrollo	Puntos estimados: 1.2
Fecha de inicio: 9 de septiembre de 2024	Fecha de fin: 10 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se configura un servicio externo de SMS para enviar notificaciones a los usuarios cuando su cita sea confirmada. El sistema debe generar el mensaje de acuerdo con el formato establecido e incluir la fecha y la dirección de la cita.	

Tabla 3.10. Tarea de ingeniería # 10

Tarea	
Número de tarea: 10	Número de Historia de usuario: 7
Nombre de la tarea: Implementar la generación del mensaje de notificación	
Tipo de tarea: Desarrollo	Puntos estimados: 0.8
Fecha de inicio: 10 de septiembre de 2024	Fecha de fin: 11 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: El sistema genera un mensaje dinámico que incluya la fecha y la dirección de la cita del cliente. El mensaje debe seguir el siguiente formato: "Su cita ha sido programada para (fecha de la cita) en (dirección de la entidad)."	

3.3.3. Tareas de ingeniería para la Iteración III

Tabla 3.11. Tarea de ingeniería # 11

Tarea	
Número de tarea: 11	Número de Historia de usuario: 8
Nombre de la tarea: Diseño e implementación del formulario de creación de trámite	
Tipo de tarea: Desarrollo	Puntos estimados: 0.8
Fecha de inicio: 11 de septiembre de 2024	Fecha de fin: 12 de septiembre de 2024

Continúa en la próxima página

Tabla 3.11. Continuación de la página anterior

Programador responsable: Téc. Carlos Brayan Rámila Chorens
Descripción: Se desarrolla un formulario en la interfaz para que el Administrador pueda crear un nuevo trámite. El formulario debe incluir los campos para el Código, Descripción y Precio.

Tabla 3.12. Tarea de ingeniería # 12

Tarea	
Número de tarea: 12	Número de Historia de usuario: 8
Nombre de la tarea: Validación de datos del formulario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 12 de septiembre de 2024	Fecha de fin: 13 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se implementan validaciones para asegurar que los datos ingresados en el formulario sean correctos. El campo Código debe ser único, el campo Precio debe ser numérico y la Descripción no debe estar vacía. Las validaciones se realizarán tanto en el frontend como en el backend.	

Tabla 3.13. Tarea de ingeniería # 13

Tarea	
Número de tarea: 13	Número de Historia de usuario: 8
Nombre de la tarea: Almacenamiento de trámite en la base de datos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.7
Fecha de inicio: 13 de septiembre de 2024	Fecha de fin: 14 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: se implementa la lógica para que, al enviar el formulario, el trámite creado se almacene correctamente en la base de datos. El sistema debe garantizar que no se repitan códigos de trámite y que toda la información esté disponible para futuras consultas y modificaciones.	

Tabla 3.14. Tarea de ingeniería # 14

Tarea	
Número de tarea: 14	Número de Historia de usuario: 10
Nombre de la tarea: Implementar la funcionalidad de eliminación de trámite	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 14 de septiembre de 2024	Fecha de fin: 15 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	

Continúa en la próxima página

Tabla 3.14. Continuación de la página anterior

Descripción: Se implementa la funcionalidad para que los usuarios con permisos administrativos puedan eliminar un trámite existente. El sistema debe presentar un mensaje de advertencia al usuario, advirtiéndole que la acción es irreversible, y solo procederá con la eliminación si el usuario confirma.
--

Tabla 3.15. Tarea de ingeniería # 15

Tarea	
Número de tarea: 15	Número de Historia de usuario: 10
Nombre de la tarea: Desarrollar la interfaz de advertencia de eliminación	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 15 de septiembre de 2024	Fecha de fin: 16 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se diseña un modal emergente en el que se muestre un mensaje de advertencia al usuario antes de proceder con la eliminación del trámite. El mensaje debe ser claro y advertir que la eliminación no puede revertirse, ofreciendo las opciones de Confirmar o Cancelar.	

3.3.4. Tareas de ingeniería para la Iteración IV

Tabla 3.16. Tarea de ingeniería # 16

Tarea	
Número de tarea: 16	Número de Historia de usuario: 15
Nombre de la tarea: Implementar el formulario de solicitud de cita	
Tipo de tarea: Desarrollo	Puntos estimados: 0.8
Fecha de inicio: 16 de septiembre de 2024	Fecha de fin: 17 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se implementa un formulario en la interfaz de usuario para permitir que el cliente solicite una cita con el notario. El formulario incluye los siguientes campos: <ul style="list-style-type: none"> • Nombre • Apellidos • Dirección Particular • Carnet de Identidad (CI) • Teléfono de localización • Motivo por el cual solicita la cita 	

Tabla 3.17. Tarea de ingeniería # 17

Tarea	
Número de tarea: 17	Número de Historia de usuario: 15
Nombre de la tarea: Validación de datos del formulario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 17 de septiembre de 2024	Fecha de fin: 18 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se implementan validaciones en el frontend y el backend para asegurar que los datos ingresados en el formulario sean correctos. El campo Carnet de Identidad debe tener el formato adecuado, el número de teléfono debe contener dígitos válidos, y el motivo de la cita debe ser obligatorio.	

Tabla 3.18. Tarea de ingeniería # 18

Tarea	
Número de tarea: 18	Número de Historia de usuario: 15
Nombre de la tarea: Almacenamiento de la solicitud de cita	
Tipo de tarea: Desarrollo	Puntos estimados: 0.7
Fecha de inicio: 18 de septiembre de 2024	Fecha de fin: 19 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se desarrolla la lógica necesaria para almacenar los datos de la cita solicitada en la base de datos. Se debe garantizar que los datos estén correctamente estructurados y que el cliente pueda ver su solicitud en su panel de usuario, con la posibilidad de seguimiento.	

Tabla 3.19. Tarea de ingeniería # 19

Tarea	
Número de tarea: 19	Número de Historia de usuario: 16
Nombre de la tarea: Implementar la opción de aceptar cita	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 19 de septiembre de 2024	Fecha de fin: 20 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se implementa una opción en la interfaz para que el notario pueda aceptar una cita pendiente. Al seleccionar esta opción, se muestra un mensaje de advertencia para confirmar la acción antes de proceder. El sistema debe notificar al cliente una vez que la cita sea aceptada por el notario.	

Tabla 3.20. Tarea de ingeniería # 20

Tarea	
Número de tarea: 20	Número de Historia de usuario: 16
Nombre de la tarea: Desarrollar la interfaz de advertencia	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 20 de septiembre de 2024	Fecha de fin: 21 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se desarrolla un modal emergente en el que se muestre el mensaje de advertencia al notario antes de confirmar la aceptación de la cita. El mensaje debe advertir claramente que la acción será irreversible y que el cliente será notificado.	

Tabla 3.21. Tarea de ingeniería # 21

Tarea	
Número de tarea: 21	Número de Historia de usuario: 16
Nombre de la tarea: Enviar notificación al cliente tras aceptación de cita	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 21 de septiembre de 2024	Fecha de fin: 22 de septiembre de 2024
Programador responsable: Téc. Carlos Brayan Rámila Chorens	
Descripción: Se implementa la lógica para enviar una notificación al cliente cuando el notario acepte la cita. Esta notificación se envía automáticamente, confirmando al cliente la cita aceptada y proporcionando los detalles correspondientes (fecha, hora y lugar).	

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 (Escribano, 2002).

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

Con el desarrollo de la presente investigación se arriba a la siguiente conclusión:

- La implementación de los módulos propuestos incorpora funcionalidades asociadas con la gestión de servicios de Transconsul S.A., permitiendo mejorar la imagen y la efectividad con el público.

Recomendaciones

A partir de los resultados obtenidos se recomienda:

- Crear la APIs sobre este modelo de aplicaciones desarrollado en Django, permitiendo la distribución de la información para diferentes aplicaciones
- Utilizar un marco de trabajo del lado del cliente (Frontend) para consumir esas APIs
- Desarrollar una aplicación multiplataforma para facilitar el uso del sistema por parte del cliente

Referencias bibliográficas

- CLAIM S.A [online] [visitado 2024-09-18]. Disponible desde: <https://patentlawyermagazine.com/company-directory/cuba-north-america/claim-s-a/> (vid. pág. 6).
- Clio | *The Industry's #1 Legal Software* | Try it for free! [online] [visitado 2024-09-19]. Disponible desde: <https://www.clio.com/> (vid. pág. 6).
- CODESCOUTS, 2022. *Patrones GRASP, el viejo SOLID* [online] [visitado 2024-03-11]. Disponible desde: <https://www.codescouts.academy/blog/grasp/> Section: blog (vid. págs. 30, 31).
- CONABI [online] [visitado 2024-09-18]. Disponible desde: <https://conabi.cu> (vid. pág. 5).
- DANIEL PECOS MARTÍNEZ, 2024. *PostgreSQL vs. MySQL* [online] [visitado 2024-02-18]. Disponible desde: <https://danielpecos.com/documents/postgresql-vs-mysql/#AEN11> (vid. pág. 8).
- ESCRIBANO, Gerardo Fernández, 2002. Introducción a Extreme Programming. *Introducción a Extreme Programming* (vid. págs. 6, 11, 12, 45).
- GAMMA, E.; HELM, R.; JOHNSON, R. y VLISSIDES, J., 1994. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley. Addison-Wesley professional computing series. ISBN 9788131736388. Url: <https://books.google.com.cu/books?id=vqp2AQAACAAJ> (vid. pág. 31).
- GILFILLAN, I., 2003. *La Biblia de MySQL*. Anaya Multimedia, S.A. La Biblia de Series. ISBN 978-84-415-1558-1. Url: <https://books.google.de/books?id=R3T1ngEACAAJ> (vid. pág. 9).
- JOSKOWICZ, José, 2008. Reglas y prácticas en eXtreme Programming. *Universidad de Vigo*. Vol. 22 (vid. págs. 7, 12).
- RAFAEL D. HERNANDEZ, 2021. *El patrón modelo-vista-controlador: Arquitectura y frameworks explicados* [freeCodeCamp.org] [online] [visitado 2024-03-11]. Disponible desde: <https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/> (vid. pág. 30).
- SARMIENTO, Romario, 2019. *Modelo - Vista - Template* [prezi.com] [online] [visitado 2024-03-11]. Disponible desde: <https://prezi.com/nd9ydrb01kqv/modelo-vista-template/> (vid. págs. 29, 30).
- SILVA, Johannerys, 2019. *PATRONES Y BUENAS PRACTICAS: Patrones GoF* [PATRONES Y BUENAS PRACTICAS] [online] [visitado 2024-03-11]. Disponible desde: https://patronesybuenaspracticass.blogspot.com/2019/12/patrones-gof_8.html (vid. pág. 31).

Generado con L^AT_EX: 27 de septiembre de 2024: 2:15pm

Apéndices