


IT TEXTHEOS

Nº	Nombre	Apellidos	Correo
1	Alejandro	Ortega Ordoñez	aortegaordonez@uma.es
2	Eduardo Noni	Stojadinovic	edunosto@uma.es
3	Iván	Aguilar García	ivanaguilargarcia@uma.es
4	Jonathan	Ruiz Oliva	jonathanruizoliva@uma.es
5	José Francisco	Villena Paez	jfvillenapaez@uma.es
6	Miguel Ángel	Caballero Serrano	miguelangelcs22@uma.es

Nombre del grupo: TexTheos
Titulación: Informática
Grupo de la titulación: 2ºA
Grupo reducido: G1
Número de grupo: 06
Repositorio GitHub: GitHub
Espacio Trello: Trello
Memoria del proyecto:  Biblioteca Online

1. Introducción.....	3
2. Roles.....	5
3. Riesgos.....	6
4. Planificación.....	9
5. Herramientas software.....	12

1. Introducción

Nuestro proyecto estará basado en la visualización de libros de forma online y/o presencial, de forma que el cliente pueda obtener el libro desde la comodidad de su hogar, ya sea descargándolo o desde la página web, o que obtenga sus libros de forma física yendo a unos de nuestros puntos de venta.

Para formar una gran comunidad de lectores, nuestros clientes podrán reunirse si así lo desean en ciertos puntos de encuentro para leer o discutir sobre sus libros favoritos. Dichas reuniones se concordarán en la web realizando una reserva, aunque siempre habrá lectores que quieran compartir sus experiencias sobre su historial de lectura.

Nuestra página web tendrá la posibilidad de creación de usuarios para marcar sus favoritos, creación de listas, escribir reseñas y la posibilidad de encontrar nuevos usuarios para compartir libros o listas. Además podrán realizar una reserva para obtener los libros de forma física más tarde, asimismo estará la posibilidad de reservar plazas para hacer una reunión. El usuario podrá suscribirse a nuestro plan para obtener mejor calidad de los libros así como mayor gama de libros, tanto más antiguos como más actuales. Desde la página web, el usuario podrá ver una muestra del libro antes de comprarlo. Al igual que los usuarios puedan escribir reseñas, existirán algunos usuarios con prioridad en sus reseñas, ya que estos serán clientes que hayan consumido una gran cantidad de libros y que serán sometidos a un pequeño examen para que puedan obtener dichos privilegios, además contrataremos expertos para que hagan sus propias reseñas.

También obtendremos información de éxito y ventas sobre los libros para que el cliente tenga la máxima información sobre el producto. Nuestra página web recopilará datos no personales del usuario para brindarle un catálogo de libros lo más adecuado a sus necesidades y gustos.

Nuestros usuarios tendrán la posibilidad de contratar de menús u obtener descuentos por puntos a través de métodos como:

- Programa de lealtad: Los clientes pueden acumular puntos por cada libro que alquilen o compren, y así poder canjearlos en descuentos o productos.
- Descuentos por volumen: Ofrecer descuentos a los clientes que alquilen o compren una gran cantidad de libros a la vez.

- Descuentos por invitación: Los clientes reciben un descuento cuando invitan a un amigo que se registra y realiza una compra.
- Descuentos de temporada: Ofrecer descuentos en ciertos momentos del año, como durante las vacaciones o el inicio del año escolar.
- Descuentos para estudiantes: Los estudiantes reciben un descuento al presentar una identificación válida.
- Cupones: Los clientes pueden recibir cupones para descuentos en futuras compras.
- Descuentos para miembros: Los clientes que pagan una cuota de membresía anual reciben descuentos en todas sus compras.
- Funcionalidad de chat en vivo o soporte al cliente para resolver dudas o problemas de los usuarios.

2. Roles

- **Líderes del Proyecto (Miguel Angel C.S. y Eduardo N.S.):** Serán responsables de la planificación general del proyecto, asignación de tareas, seguimiento del progreso y comunicación con el equipo. También tomarán decisiones sobre el desarrollo del software asegurándose de que se cumplan los objetivos propuestos y que se implementen las ideas propuestas.

- **Analistas de Requisitos (Miguel Angel C.S. y José F.V.P.):** Se encargarán de recopilar, analizar y documentar los requisitos del cliente. Junto con el equipo de desarrollo garantizarán que el producto final cumpla con las expectativas del cliente.

- **Users Interface (Jonathan R.O. y Alejandro O.O.):** Diseñarán la interfaz de usuario del sistema, asegurándose de que sea intuitiva y fácil de usar. Colaborarán mayoritariamente con los analistas de requisitos y programadores para garantizar una buena experiencia de usuario.

- **Programadores (Jonathan R.O., Eduardo N.S. y Iván A.G.):** Desarrollarán el software siguiendo los requisitos y el diseño establecido. Serán responsables de escribir código limpio y bien documentado.

- **Testers (José F.V.P., Alejandro O.O., Iván A.G.):** Realizarán pruebas exhaustivas del software para identificar y reportar cualquier error o defecto. Junto con los programadores solucionarán los problemas detectados y garantizarán la mayor calidad posible del software final.

- **Project Manager (Miguel Angel C.S.):** Supervisará el progreso del proyecto, gestionará los recursos y resolverá cualquier problema que surja durante el desarrollo. Se asegurará de que el proyecto se entregue a tiempo.

- **Representante del Cliente (Eduardo N.S.):** Actuará de intermediario entre el equipo de desarrollo y el cliente, transmitiendo sus necesidades y expectativas al equipo y proporcionando retroalimentación sobre el progreso del proyecto.

3. Riesgos

1. Riesgo Tecnológico

Descripción: Problemas que podrían surgir durante la integración de las diferentes tecnologías utilizadas en el desarrollo del proyecto, como incompatibilidades entre herramientas, problemas de configuración o dificultades en la implementación del sistema.

Probabilidad: Moderada, dado que el proyecto implica el uso de múltiples herramientas y tecnologías.

Efectos: Serios, ya que podría retrasar significativamente el desarrollo del proyecto y afectar la calidad del software final.

Estrategia para mitigarlo: Los programadores y testers deberán de realizar pruebas de forma regular durante todo el ciclo de desarrollo y resolver cualquier problema de manera eficiente.

2. Cambios en los Requisitos

Descripción: Solicitud por parte del cliente de cambios frecuentes en los requisitos del proyecto. Pueden surgir debido a una comprensión incompleta de sus restricciones o a cambios en sus necesidades.

Probabilidad: Alta, ya que es común que los requisitos evolucionen a lo largo del tiempo a medida que el cliente adquiere una mejor comprensión de sus necesidades.

Efectos: Serios, dado que pueden afectar a la planificación del proyecto, aumentar los costes y el tiempo de desarrollo, y comprometer la calidad del software final.

Estrategia para mitigarlo: Establecer una comunicación clara y abierta con el cliente desde el principio del proyecto mediante una metodología ágil que

permita adaptarse rápidamente a los cambios y mantener un proceso de desarrollo que facilite la incorporación de nuevos requisitos.

3. Rendimiento del Sistema

Descripción: El sistema no puede soportar la carga esperada de usuarios o no cumple con los requisitos de rendimiento establecidos.

Probabilidad: Moderada, ya que el rendimiento puede verse afectado por factores como el tamaño de la base de datos, la complejidad de las consultas o la infraestructura de red.

Efectos: Catastróficos, debido a que podría resultar en una mala experiencia para los usuarios, pérdida de clientes y daño a la reputación de la biblioteca.

Estrategia para mitigarlo: El equipo de desarrollo realizará pruebas exhaustivas de rendimiento durante todo el ciclo de desarrollo identificando y resolviendo los cuellos de botella de rendimiento de manera constante, optimizando el código e identificar cualquier problema de rendimiento en tiempo real.

4. Riesgo de Seguridad de Datos

Descripción: La seguridad de los datos del sistema se ve comprometida debido a vulnerabilidades en el software o a acciones de terceros.

Probabilidad: Alta y las amenazas a la seguridad pueden provenir de una variedad de fuentes, incluidos hackers, malware y errores humanos.

Efectos: Catastróficos, ya que la pérdida de datos sensibles puede tener graves consecuencias legales y financieras para la biblioteca, así como dañar su reputación y la confianza de los usuarios.

Estrategia para mitigarlo: El equipo de desarrollo implementará prácticas de seguridad sólidas, como encriptación de datos, autenticación de usuarios y control de acceso. Además, se realizarán parches de seguridad de manera regular.

5. Riesgo de Recursos Insuficientes

Descripción: El proyecto no cuenta con los recursos adecuados, como personal, tiempo o presupuesto, para completarse con éxito.

Probabilidad: Moderada, debido a que la disponibilidad de recursos puede verse afectada por factores como cambios en la disponibilidad del personal o restricciones en el presupuesto.

Efectos: Serios, ya que la falta de recursos puede llevar a retrasos en el proyecto, aumento de los costes y afectar a la calidad del software final.

Estrategia para mitigarlo: Planificación cuidadosa de los recursos desde el principio del proyecto, identificando claramente los recursos necesarios y serán asignados de manera eficiente. Además, deberemos de mantener una comunicación abierta con todas las partes implicadas para gestionar expectativas y resolver cualquier problema de recursos.

6. Riesgo de Cambios en la Tecnología

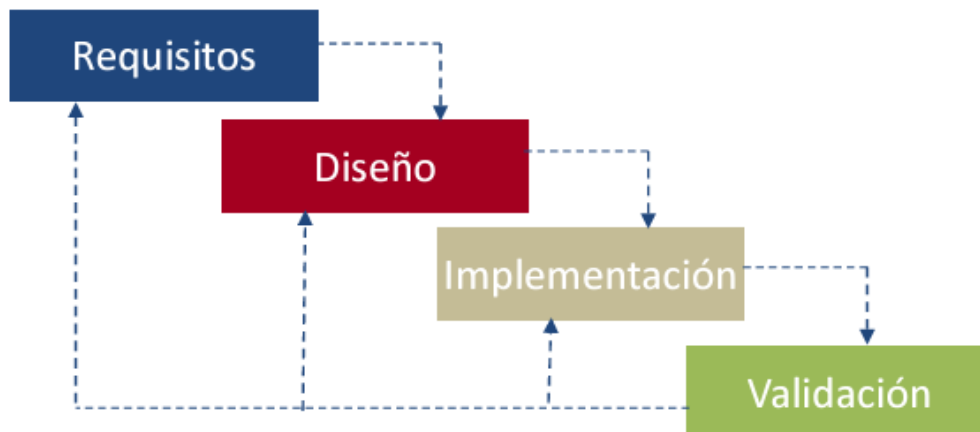
Descripción: Cambios significativos en la tecnología o en las prácticas de desarrollo durante el transcurso del proyecto.

Probabilidad: Moderada, ya que las tecnologías evolucionan constantemente, y es posible que surjan nuevos enfoques que nos puedan afectar al desarrollo del proyecto y nos interese cambiar.

Efectos: Tolerables/Serio, dado que si se producen cambios en la tecnología surgirá la necesidad de adaptarse a las nuevas necesidades o entornos de programación y requerirá conocimiento previo que puede que no poseamos inicialmente. Sin embargo, estos efectos pueden gestionarse con una planificación adecuada.

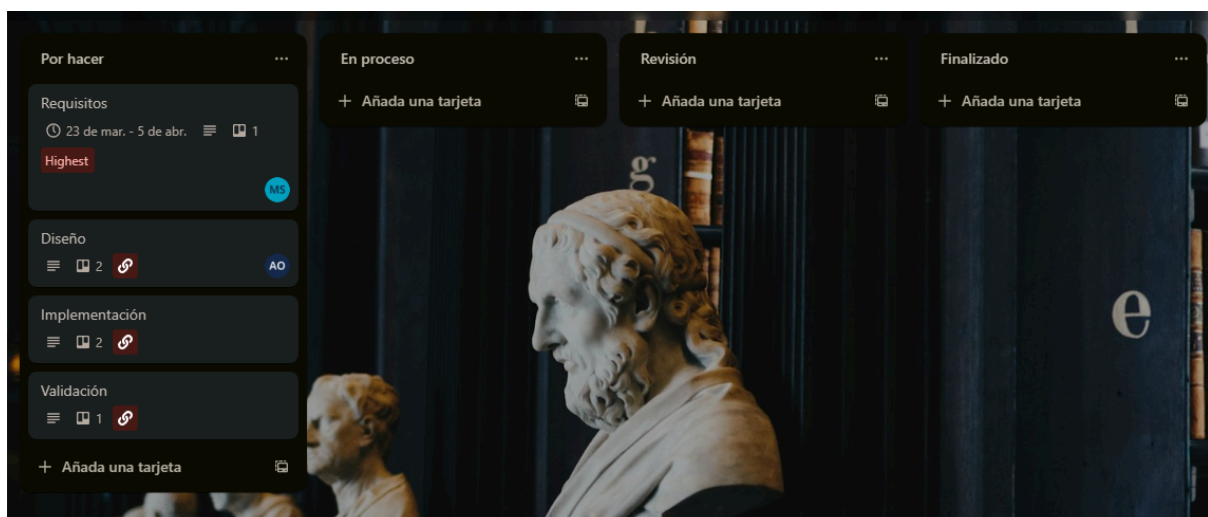
Estrategia para mitigarlo: El equipo de desarrollo se mantendrá actualizado sobre las tendencias tecnológicas y las mejores prácticas actuales y estará preparado para adaptarse a cambios si es necesario.

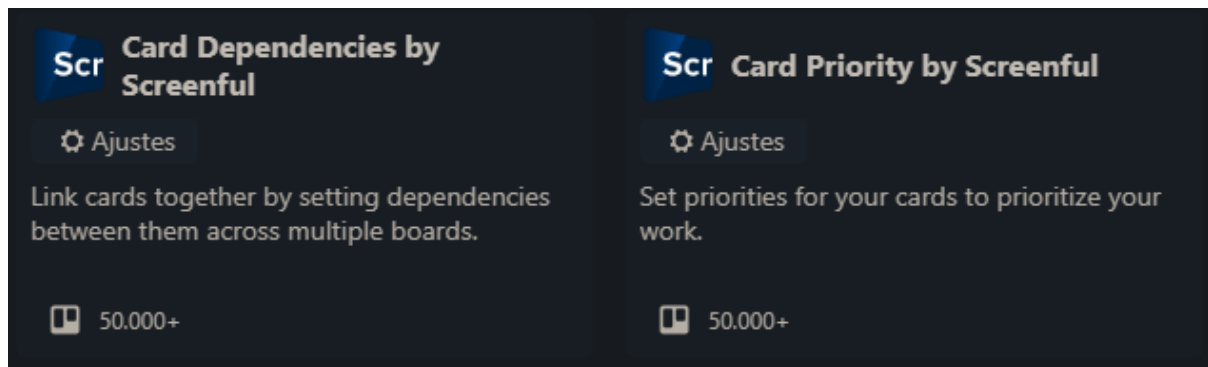
4. Planificación



El modelo en cascada, es la propuesta de un enfoque metodológico para hacer un software que consiste en ordenar de forma lineal las distintas etapas que debes de seguir al momento de desarrollar tu software. Este modelo ha sido seleccionado por nosotros debido a que es un modelo muy sencillo para aprender a hacer software. Otro motivo por el que lo hemos elegido es porque al analizar las desventajas que se nos pueden presentar por utilizar este modelo no nos son demasiado problemáticas dada la situación del proyecto, entre algunas desventajas del proyecto encontramos:

- El cliente a veces no conoce todos los requisitos al inicio.
- Los miembros del equipo deben esperar a que terminen tareas precedentes.
- El cliente debe tener paciencia.
- Poco flexible.
- Los proyectos reales raramente siguen el flujo secuencial del modelo.





- Requisitos:

Los requisitos son las especificaciones y características que el sistema debe cumplir para satisfacer las necesidades y requerimientos del cliente. Se dividen principalmente en requisitos funcionales, es decir, son los requisitos que describen las funciones y características específicas que el sistema debe realizar. Incluyen acciones que el sistema debe permitir a los usuarios realizar o procesos que debe automatizar.

En cambio, los no funcionales son los requisitos que describen las características de calidad y restricciones de rendimiento que el sistema debe cumplir. No se refieren a una función específica del sistema, sino a cómo debe funcionar el sistema en términos de rendimiento, seguridad, usabilidad, etc.

- Diseño:

#Estructura:

- Página de inicio/registro de sesión: El cliente registra un usuario con su contraseña para iniciar sesión y poder acceder más tarde. Debe tener un usuario único.
- Buscador: Los usuarios pueden buscar el libro por título, escritor, año y categoría.
- Categorías: Los libros se ordenan por categorías y etiquetas.
- Creación de etiquetas y favoritos: El usuario puede crear y guardar etiquetas, además de la etiqueta favoritos que viene por defecto
- Foro de discusiones en la pestaña del libro: Los usuarios pueden comentar los libros que han leído.

- Comentarios dentro de la comunidad: Los usuarios pueden interactuar entre sí dentro de nuestro foro.
- Opción de compra: Los usuarios pueden descargar los libros después de comprarlos.
- Alquilar: Los usuarios pueden alquilar el libro durante cierto tiempo para leerlo por un precio moderado.

#Cómo se desarrollan los elementos:

El programa inicia con el inicio de sesión/registro, es intermediario entre los usuarios y el programa para que la gente no pueda entrar en usuarios ajenos; también las demás opciones que te da el programa.

Dentro de la página principal tenemos el buscador que se relaciona directamente con los libros, también tenemos las configuraciones de la cuenta, las categorías que se relaciona con los libros y la sección del foro de la comunidad que se relaciona con otros usuarios, además de la sección de compra y alquiler de libros. En la sección de libros existe relación con una sección de relacionados, guardar en alguna categoría, comentarios sobre el libro por parte de otros usuarios además de opción para comprar o alquilar.

- Implementación:

Utilizaremos para crear nuestro software dos elementos fundamentales los cuales son los siguientes:

- Lenguaje de programación: Java
- IDE: IntelliJ

La razón por la que vamos a usar estas dos es simple. La primera es el lenguaje de Java porque este es el lenguaje de programación que mejor manejan nuestros programadores y es un buen lenguaje para aplicaciones de escritorio. La segunda que usaremos es IntelliJ ya que este es un IDE que los programadores también manejan con agilidad, también permite la capacidad de implementar una interfaz gráfica.

- Validación:

En la fase de validación comprobaremos el funcionamiento adecuado del software, veremos si cumple todos los requisitos propuestos por el cliente y controlaremos la correcta realización de las tareas anteriores. Para ello ejecutaremos tests y dejaremos que el cliente pruebe las funcionalidades de la aplicación.

5. Herramientas software

Comunicación		
	Trello	
	Discord	
	WhatsApp	
	Miro	
Desarrollo		
	IntelliJ	
	GitHub	
	GitKraken / GitHub Desktop	