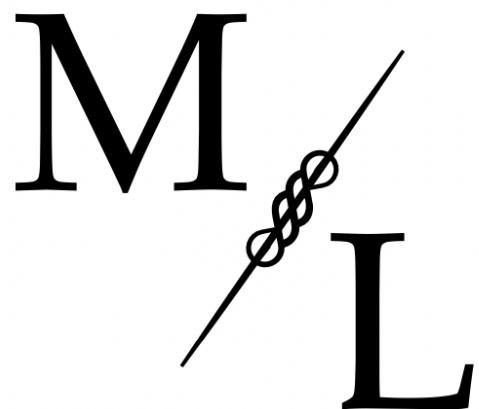


MynimaList



MynimaList

Tabla de contenidos

Tabla de contenidos	2
Integrantes del grupo	3
Introducción	4
Roles	5
Gestión de riesgos	6
Tipo de riesgo: Personal	6
Tipo de Riesgo: Requisitos	6
Tipo de Riesgo: Estimación	6
Tipo de Riesgo: Organización	6
Tipo de Riesgo: Tecnológico	6
Planificación	7
Modelo de proceso software elegido: Scrum	7
Tableros Trello	8
Herramienta de comunicación	11
Herramienta de trabajo colaborativo	11
Herramienta de elaboración de documentos	11
Requisitos	12
Funcionales	12
No funcionales	12

Integrantes del grupo

- Jacobo Elisha Garrucho (jacoboeg@uma.es)
- Julia Pérez Barreales (juliaperezb@uma.es)
- Álvaro Sánchez Hernández (alvaro.s.h@uma.es)
- Jesús Escudero Moreno (xexu65@uma.es)
- Isidro Javier García Fernández (isidrojova@uma.es)
- Juan Manuel García Delgado (juanma01@uma.es)
- José Antonio Luque Salguero (joseantonioluquesalguero1212@uma.es)
- David Ramírez Palacios (ramirezpalaciosd@uma.es)

Enlace repositorio GitHub: <https://github.com/Jacobo-EG/g10-MynimaList>

Introducción

MynimaList consiste en una aplicación web minimalista e intuitiva para realizar anotaciones colaborativas como listas de tareas, ideas de proyectos o recordatorios.

Buscamos crear un espacio en el que se puedan gestionar distintas listas, tanto privadas, como compartidas entre varios usuarios que tengan necesidades comunes.

La idea inicial es que un usuario pueda crear una lista de tareas o con anotaciones en la que posteriormente se puedan añadir más editores si se trata de una lista compartida.

Nos decidimos por esta idea ya que, en nuestro grupo de estudio echamos en falta una forma sencilla de poder escribir en tiempo real las tareas que necesitamos realizar y poder compartir las con el resto de nuestros compañeros.

De esta manera, se facilita la organización y permite mantener informados a los integrantes del grupo en todo momento.

Roles

- Álvaro Sánchez Hernández: tester, implementador
- David Ramírez Palacios: Scrum Master, implementador
- Isidro Javier García Fernández: diseñador, Product Owner
- Jacobo Elichá Garrucho: diseñador, Scrum Master
- Jesús Escudero Moreno: Product Owner, tester
- José Antonio Luque Salguero: implementador, tester.
- Juan Manuel García Delgado: tester, implementador
- Julia Pérez Barreales: diseñadora, implementadora

Gestión de riesgos

- Tipo de riesgo: **Personal**
 - Descripción del riesgo: Personal clave está enfermo o no está disponible en momentos críticos.
 - Probabilidad: moderada.
 - Efectos del riesgo: Serio.
 - Estrategia para mitigarlo: Reorganizar el equipo de desarrollo para que haya más superposición de trabajo y, por lo tanto, el personal conozca el trabajo de otros compañeros.
- Tipo de Riesgo: **Requisitos**
 - Descripción del riesgo: Se proponen unos cambios de requisitos que necesitan un importante rediseño.
 - Probabilidad: Baja.
 - Efecto del riesgo: Serio.
 - Estrategia para mitigarlo: Obtener la información de trazabilidad para evaluar el impacto de los cambios en los requisitos, Maximizar la ocultación de la información en el diseño.
- Tipo de Riesgo: **Estimación**
 - Descripción del riesgo: Se subestima el tiempo necesario para el desarrollo del software.
 - Probabilidad: Alto.
 - Efecto del riesgo: Serio.
 - Estrategia para mitigarlo: Realizar una estimación por exceso.
- Tipo de Riesgo: **Organización**
 - Descripción del riesgo: Una reestructuración de la organización provoca un cambio de los gestores responsables del proyecto.
 - Probabilidad: Bajo.
 - Efecto del riesgo: Tolerable.
 - Estrategia para mitigarlo: Preparar un documento informativo que muestre la forma en la cuál el proyecto realiza una importante contribución a los objetivos.
- Tipo de Riesgo: **Tecnológico**
 - Descripción del riesgo: La base de datos usada en el sistema no es capaz de procesar tantas transiciones por segundo como se esperaba.
 - Probabilidad: Moderado.
 - Efecto del riesgo: Serio.
 - Estrategia para mitigarlo: Investigar la posibilidad de comprar una base de datos de alto rendimiento.

Planificación

Modelo de proceso software elegido: Scrum

Hemos elegido este modelo de trabajo, ya que fomenta la comunicación entre los miembros del grupo, conocimiento de las potencialidades, optimización del tiempo y adaptabilidad a cambios.

Mediante este modelo de proceso software podemos llevar a cabo un consenso de opiniones e ideas acerca de cómo desarrollar las distintas partes del proyecto o cómo resolver los problemas que se puedan presentar.

Pensamos que es buena idea utilizar este método, pues en un principio nuestro proyecto parte de una cierta incertidumbre.

Vamos a intentar realizar ciclos cortos de Sprint para intentar reducir los posibles riesgos que puedan suceder a lo largo del desarrollo de nuestra aplicación web.

Para el desarrollo de MynimaList mediante el modelo Scrum deberemos implementar los **documentos** propios de dicha metodología de trabajo, como lo son:

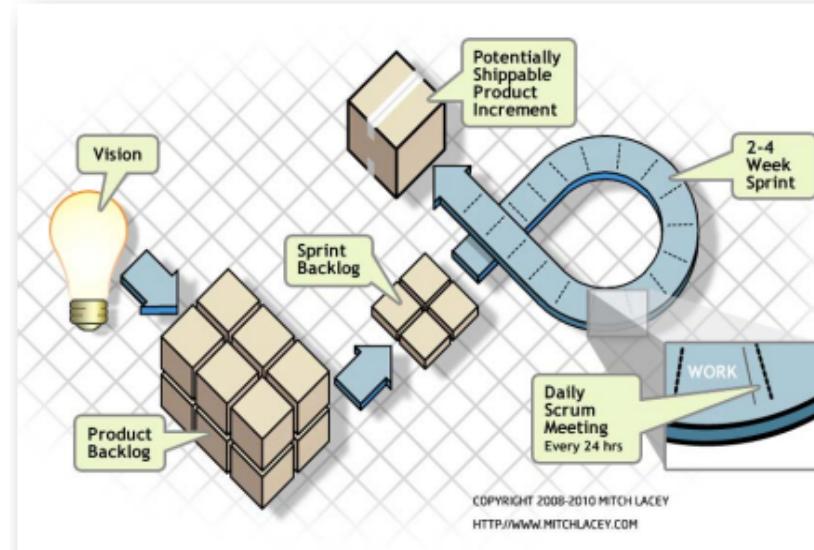
- Product Backlog → reunir los requisitos del proyecto y describir las funcionalidades
- Sprint Backlog → requisitos que se desarrollaran en el siguiente sprint
- Burndown Chart → gráfica que mide la cantidad de requisitos en el Backlog

También hemos seguido la división de cada uno de los miembros del equipo en distintos **roles** que se utilizan en el método Scrum. Como hemos mencionado anteriormente, nos hemos asignado roles como:

- Producto Owner
- Scrum Master
- Equipo de desarrollo: diseñador, implementador, tester

También llevaremos a cabo **reuniones** correspondientes al modelo de proceso software Scrum, con el objetivo de establecer la situación actual del proyecto, problemas que se nos presenten y posibles soluciones a los mismos. Desarrollaremos documentos como:

- Daily Scrum / Stand-up meeting → reunión sobre el estado del proyecto
- Sprint Planning → establecer qué trabajo se hará
- Sprint Review Meeting → Inspeccionar el trabajo realizado
- Sprint Retrospective → Análisis del propio sprint



Tableros Trello

Por hacer

Entrega 27 marzo

⌚ ⏰ 26 de mar. ⚙️

DP I JG JM JS JD JB AH

Elección software

⌚ ⏰ 26 de mar. ⚙️

DP I JG JM JS JD JB AH

Sprint planning

⌚ ⚙️

DP I JG JM JS JD JB AH

+ Añada una tarjeta

Entrega 27 marzo

en la lista Por hacer ⚙️

Miembros

AH DP I JG JM JS JD JB +

Vencimiento

26 de mar. a las 23:59 ▾

Descripción Editar

Incluir requisitos en el PDF, con los requisitos detallados en Magicdraw.

Elección software

en la lista Por hacer ⚙️

Miembros

AH DP I JG JM JS JD JB +

Vencimiento

26 de mar. a las 23:59 ▾

Descripción Editar

Elegir software con el que se trabajará

Sprint planning

en la lista Por hacer ⚙️

Miembros

AH DP I JG JM JS JD JB +

Descripción Editar

Reunión Sprint planning.

Trabajando

...

Desarrollo Product Backlog (Requisitos)

⌚ 19 de mar. ⚙

DP I JG JM JS JD JB AH

Entrega 20 marzo

⌚ 19 de mar. ⚙

DP I JG JM JS JD JB AH

+ Añada una tarjeta

...

Desarrollo Product Backlog (Requisitos)

en la lista Trabajando ⓘ

Miembros

AH DP I JG JM JS JD JB +

Vencimiento

hoy a las 23:59 **vence pronto** ↗

Descripción Editar

Elaborar el product backlog, previo al sprint planning.

Entrega 20 marzo

en la lista Trabajando ⓘ

Miembros

AH DP I JG JM JS JD JB +

Vencimiento

hoy a las 23:59 **vence pronto** ↗

Descripción Editar

Se entrega una primera planificación de las tareas que hemos decidido que se deben llevar a cabo. Incluimos el product backlog en el PDF que entregamos.

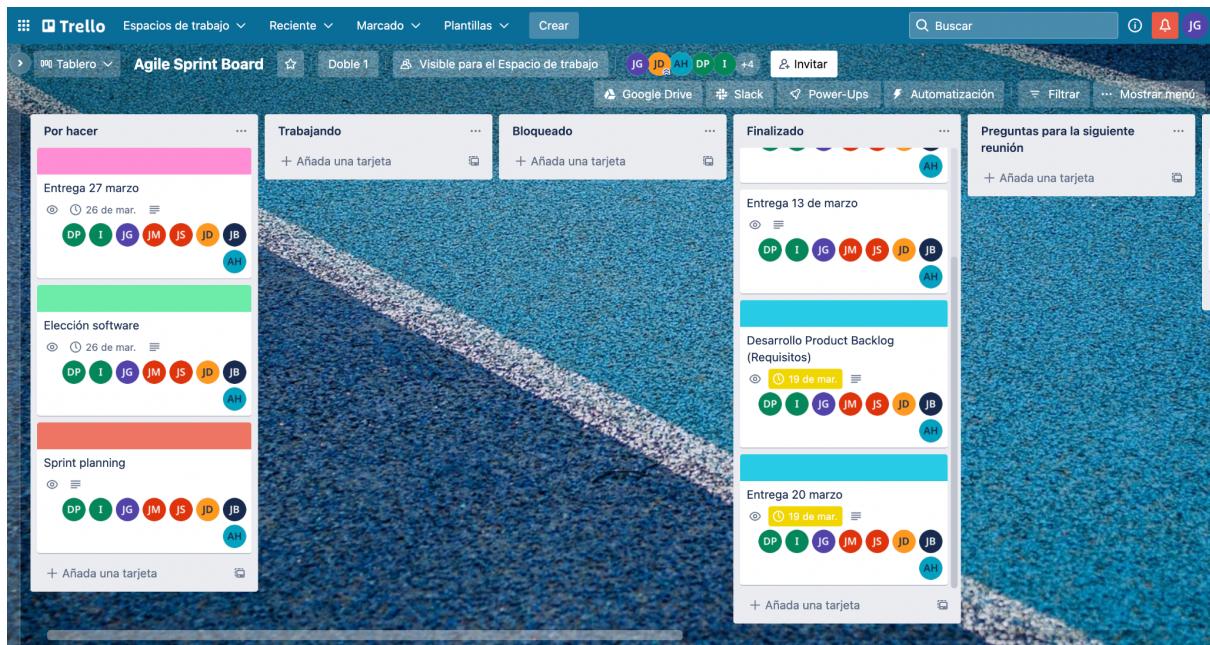
Bloqueado

...

+ Añada una tarjeta

...

Y después de modificar este documento para el 20 de marzo:



Herramientas software

Herramientas software usadas: grupo de Whatsapp, Github, Trello, Google Docs, Figma, Discord.

- **Herramienta de comunicación**

Las aplicaciones elegidas para la comunicación entre los integrantes del grupo son Whatsapp, donde hemos creado un grupo para coordinarnos, y Discord, donde hemos creado un servidor para poder realizar reuniones a través de videollamadas.

- **Herramienta de trabajo colaborativo**

Las herramientas en línea seleccionadas para la realización del proyecto son Github, donde se ha creado un repositorio al que se irán subiendo los documentos relacionados con el proyecto; Trello, en el que hemos hecho un espacio de trabajo donde registramos las tareas *por hacer, en proceso y hechas* mediante tarjetas; y Figma, que utilizaremos para probar distintos diseños hasta encontrar el más adecuado para nuestro proyecto.

- **Herramienta de elaboración de documentos**

Para la realización de documentos se ha elegido Google Docs, ya que nos permite realizar cambios al mismo tiempo en un mismo documento.

Requisitos

Funcionales

1. Página de inicio
2. Menú de listas
 - 2.1. Título de listas
 - 2.2. Ordenar las listas con propio criterio
3. Botón para crear listas
4. Guardar lista
 - 4.1. Guardado con botón
 - 4.2. Autoguardado
5. Listas colaborativas
6. Registro de usuarios (e inicio de sesión)
7. Cerrar sesión
8. Entrar a listas colaborativas mediante un enlace (sin usuario)
9. Tareas con fecha de entrega
10. Botón para crear enlace a lista colaborativa
11. Estilos de letras (tipografía)
12. Modo oscuro
13. Máximo de creación de listas diario: 10 listas diarias como máximo

No funcionales

1. Autoguardados cada cierto tiempo
2. Encriptado de contraseñas
3. La información de las tareas, usuarios, contraseñas... se almacena en nuestra base de datos
4. Página intuitiva y sencilla
5. Licencia del software