



Nombre del grupo:	Wise Wallet
Titulación:	Ingeniería del software
Grupo de la titulación (A/B/C/D/...):	A
Número de grupo, dentro del grupo reducido:	GR1-03
Repositorio Github (url):	<a href="https://github.com/franra18/WiseWallet.git">https://github.com/franra18/WiseWallet.git</a>
Espacio Trello (url):	<a href="https://trello.com/b/KsjOFd44">https://trello.com/b/KsjOFd44</a>
Memoria del proyecto (Google Docs url):	<a href="https://docs.google.com/document/d/1yeXdGqyciWbo2FDvcBGJKMdxkrRT2QkFJecdUtmZAVA/edit?usp=sharing">https://docs.google.com/document/d/1yeXdGqyciWbo2FDvcBGJKMdxkrRT2QkFJecdUtmZAVA/edit?usp=sharing</a>

## MIEMBROS:

Número de miembro	Nombre	Apellidos	Correo electrónico UMA
1	Francisco	Ramírez Cañadas	franramirez@uma.es
2	Marcos	Luque Montiel	maarcoos_8@uma.es
3	Soraya	Bennai Sadqi	sorayasadqi@uma.es
4	Manuel	Espinosa Llabrés	manuespinosallabres@uma.es
5	Víctor	Gómez Díaz	vgodiaz@uma.es
6	Alberto	Banderas Sánchez	albertobs@uma.es

## TABLA DE CONTENIDOS:

***SECCIÓN 1: Introducción. Hay que explicar el problema que se intenta solucionar con el sistema software a desarrollar.***

Administrar el dinero es un aspecto importante de nuestra vida diaria. En el día a día, es muy fácil perder de vista nuestros hábitos de gasto, lo que genera problemas financieros. Una forma de evitarlo es haciendo un seguimiento de los mismos. Al mantener un registro de lo que gastamos, podemos identificar áreas en las que podemos recortar y ahorrar dinero.

Beneficios de realizar un seguimiento de sus hábitos de gasto

El seguimiento de los hábitos de gasto tiene varias ventajas que pueden ayudarle a lograr la estabilidad financiera. Aquí hay algunos beneficios:

### **1. Identificar gastos innecesarios**

El seguimiento de sus finanzas le ayudará a identificar áreas en las que está derrochando el dinero. Le permite identificar aquellos gastos que pueden reducirse, como salir a cenar con demasiada frecuencia o comprar ropa cara.

## **2. Una mejor planificación de presupuesto**

Esta herramienta le ayuda a crear un presupuesto y planificar mejor sus finanzas. Cuando tenga una idea clara de cuánto dinero entra y sale, podrá asignar mejor sus fondos, pagar deudas y ahorrar para gastos futuros.

## **3. Detectar y prevenir el fraude**

Al revisar periódicamente sus extractos bancarios y facturas de tarjetas de crédito, puede detectar cualquier actividad fraudulenta en sus cuentas, y cualquier transacción no autorizada e informarla de inmediato.

***SECCIÓN 2: Roles. Indicar los roles que tiene cada integrante del equipo. Cada uno debe tener dos roles, y cada rol debe estar cubierto, al menos, por dos compañeros. [Recordad que tener un rol significa ser responsable de esa tarea, pero todos los integrantes del equipo deben saber de todas las tareas.]***

Los diferentes roles serán: diseñador, scrum master, desarrollador, product owner, tester, analista.

Estos serán los roles que tendremos cada uno de nosotros:

- Marcos: tester y analista.
- Soraya: diseñador y scrum master.
- Fran: scrum master y tester.
- Alberto: analista y product owner.
- Manu: diseñador y desarrollador.
- Víctor: product owner y desarrollador.

**SECCIÓN 3: Gestión del Riesgo. Hay que identificar los riesgos del proyecto y, para cada uno, indicar: tipo de riesgo, descripción del riesgo, probabilidad (muy baja, baja, moderada, alta o muy alta), efectos del riesgo (catastrófico, serio, tolerable o insignificante) y estrategia para mitigarlo.**

<b>RIESGO</b>	<b>PROBABILIDAD</b>	<b>EFFECTOS</b>	<b>TIPO</b>	<b>ESTRATEGIA PARA MITIGARLO</b>
Es imposible reclutar personal con las habilidades necesarias	Muy baja	Tolerable	Personal	Formar a los integrantes actuales
Personal clave está enfermo o no está disponible en momentos críticos.	Baja	Serio	Personal	Repartir sus tareas temporalmente entre el resto de personal
Los componentes software que se iban a reutilizar tienen defectos que limitan su funcionalidad	Baja	Insignificante	Tecnología	Crear dichos componentes desde cero
Se proponen unos cambios de requisitos que necesitan un importante rediseño.	Moderada	Serio	Tecnología	Reunir a la mayoría de integrantes para trabajar en dichos cambios
Una reestructuración de la organización provoca un cambio de los gestores responsables del proyecto.	Baja	Tolerable	Organización	Informar al nuevo responsable de sus tareas
La base de datos usada en el sistema no es capaz de procesar tantas transiciones por segundo como se esperaba.	Muy baja	Serio	Tecnología	Conectar de nuevo la base de datos
Se subestima el tiempo necesario para el desarrollo del software.	Moderada	Catastrófico	Estimación	Planificar las tareas con antelación y pedir ayuda si es necesario

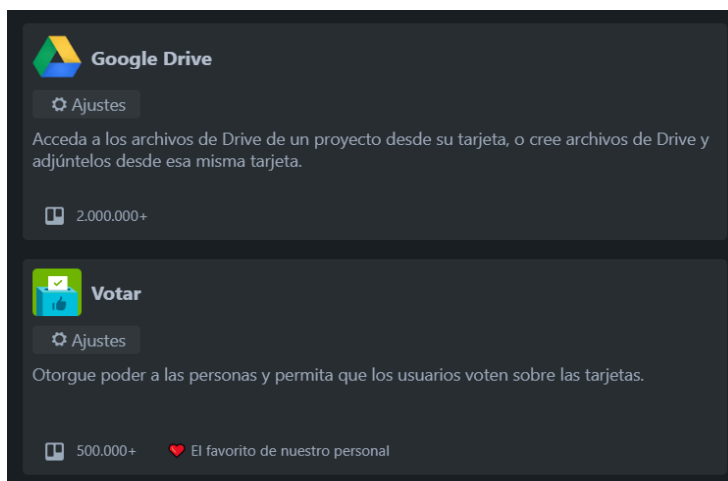
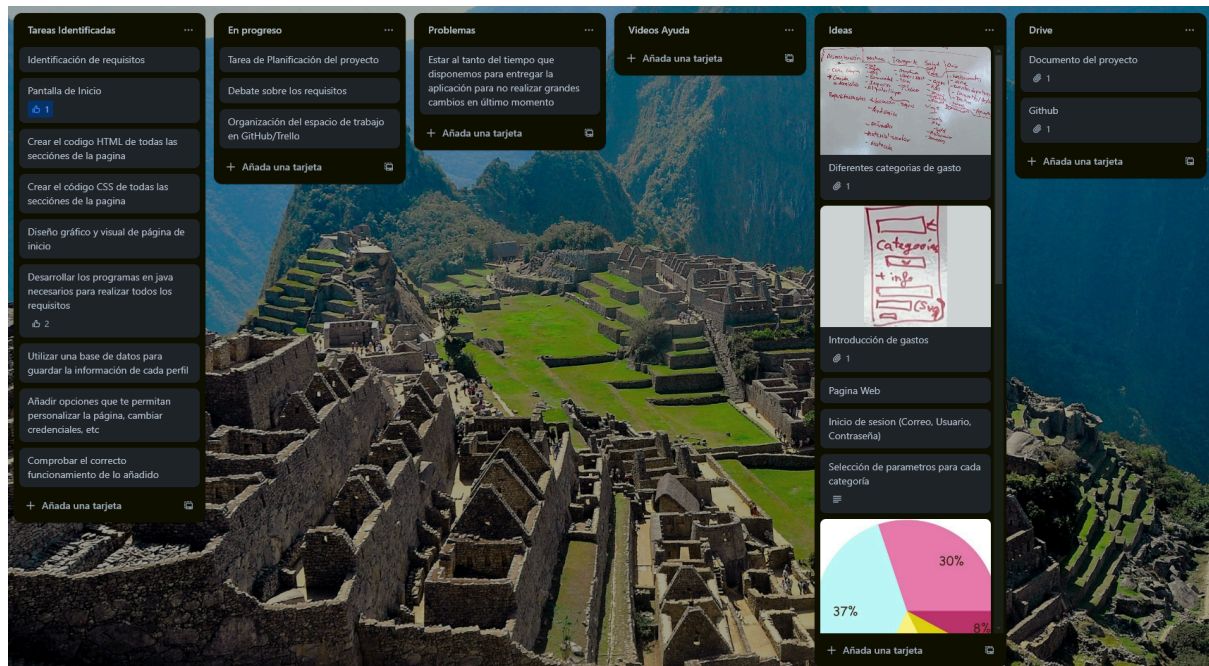
Los clientes no aprecian la magnitud de los cambios de requisitos que proponen.	Moderada	Serio	Requisitos	Proponer otras alternativas
Formación necesaria para el personal no está disponible	Muy baja	Serio	Herramientas	Asignarle otro rol
El índice de reparación de errores está subestimado	Muy baja	Catastrófico	Estimación	Reestructuración de los roles y modificación de los requisitos
El tamaño del software está subestimado.	Alta	Catastrófico	Tecnología	Intentar detectar información o código sobrante y optimizarlo
Falta de entendimiento entre los integrantes de una misma tarea.	Moderada	Tolerable	Organización	Que el scrum master haga de moderador entre ambas partes
Hay algún fallo en el sistema de guardado.	Moderada	Tolerable	Tecnología	Realizar guardados cada cierto tiempo y volver a reconstruir lo que se haya perdido
Sobrecarga de trabajo	Baja	Tolerable	Organización	Repartir las tareas de forma más dividida
Requisitos mal definidos	Moderada	Tolerable	Tecnología	Reunirse y replantear dichos requisitos
Miembros del equipo con actitudes negativas hacia el proyecto sabotean cualquier esfuerzo.	Muy baja	Tolerable	Personal	Intentar que todos los miembros del equipo estén igual de implicados en el proyecto y solucionar cualquier disputa que pueda surgir.
Rendimiento del software no deseado	Baja	Catastrófico	Tecnología	Analizar la parte ineficiente y encontrar un mejor enfoque.
Los componentes técnicos tienen vulnerabilidades de seguridad	Baja	Serio	Tecnología	Intentar no acceder a páginas extrañas o utilizar herramientas que puedan dañar nuestro proyecto.

	Insignificante	Tolerable	Serio	Catastrófico
Muy baja		- Imposible reclutar personal - Actitudes negativas	- Formación no disponible - Base de datos saturada	- Índice de reparación de errores subestimado
Baja	- Componentes reutilizados defectuosos	- Cambio de gestores - Sobrecarga de trabajo	- Personal clave no disponible - Vulnerabilidades de seguridad	- Rendimiento de software no deseado
Moderada		- Falta de entendimiento - Fallo al guardar - Requisitos mal definidos	- Cambios de requisitos - Clientes no aprecian la magnitud de los cambios	- Se subestima el tiempo necesario
Alta		- Tamaño de software subestimado		
Muy alta				

**SECCIÓN 4: Planificación.** *Se debe indicar el modelo de proceso software elegido (y por qué se ha elegido) y pegar capturas de pantalla de los tableros Trello con las tareas identificadas y su organización. Si se han usado módulos (Power-Ups), pegar también capturas de pantalla con lo que muestran. Explicad brevemente las tareas.*

El modelo en cascada es el que usaremos, ya que conocemos los diferentes requisitos que son necesarios para este proyecto. Favorable en nuestro caso, ya que nos encontramos en la fase inicial del proyecto. Se procurará en todo momento seguir el flujo secuencial establecido, dando a conocer al cliente todos los requisitos desde el comienzo y contar con una organización en las tareas asignadas, informando al resto del grupo de los avances realizados.

En cuanto a las tareas, vamos a repartir de forma equivalente el desarrollo de código tanto en HTML, CSS y Javascript. Participaremos de forma activa en la aportación de ideas para el diseño de la página, la creación de una base de datos que recoja toda la información que se utilizará y la de un perfil de usuario o casos de uso que describan acciones específicas en la página web, realización de pruebas de calidad, etc.



Estos son los Power-Ups que utilizaremos en principio.

**SECCIÓN 5: Requisitos.** *Añadir los requisitos como se explica al final de las diapositivas. Es decir, identificar con tarjetas de colores los RF y los RNF (obligatorios y opcionales) en forma de historias de usuario. Se valorará positivamente que los requisitos lleven asociadas pruebas de aceptación. Cada historia de usuario debe tener un identificador (por ejemplo, RF1) y un título. Pegar como imagen el diagrama de requisitos hecho en Visual Paradigm (en dicho diagrama no se indica el texto de las historias de usuario, solamente el identificador y título). Cada grupo debe definir como mínimo tantos RFs como 5x su número de miembros, y tantos RNFs como su número de miembros. Por ejemplo, un grupo de 6 miembros debe definir al menos 30 RFs y 6 RNFs.*

- **RF1: INICIO DE SESIÓN:** Como analista, quiero implementar un sistema de inicio de sesión seguro que requiera credenciales de usuario (nombre de usuario y contraseña) para acceder a la plataforma.

**PRUEBAS DE ACEPTACIÓN:**

- El nombre de usuario debe coincidir con uno que ya esté en la base de datos.
- Si el nombre coincide con alguno de la base de datos pero la contraseña no es la asociada, recibe un mensaje de error.

- **RF2: REGISTRO DE USUARIO:** Como analista quiero permitir a los usuarios registrarse en la plataforma proporcionando la información necesaria, como nombre, dirección de correo electrónico y contraseña para coleccionar los datos.

**PRUEBAS DE ACEPTACIÓN:**

- Todos los campos deben tener valor, en caso contrario se informará con un mensaje de error.
- El correo no debe coincidir con ningún otro correo en la base de datos, en caso contrario se informará con un mensaje de error.
- Se registra un usuario nuevo, se pide un listado de usuarios y aparece el usuario nuevo.

- **RF3: GESTIÓN DE USUARIOS:** Como desarrollador quiero proporcionar funcionalidades para que los usuarios administren sus cuentas como cambiar la contraseña, actualizar la información de perfil, cerrar sesión, eliminar la cuenta, etc.

**PRUEBAS DE ACEPTACIÓN:**

- Se modifican los datos de un usuario, se pide un listado de usuarios y aparece el usuario con los datos modificados.
- Se elimina un usuario, se pide un listado de usuarios y aparece el cliente como eliminado.



- **RF4: SUBIDA DE GASTOS:** Como desarrollador quiero proporcionar una interfaz fácil de usar para que los usuarios ingresen información sobre sus gastos. Incluye campos como la fecha, el monto, la descripción y el tipo de gasto.

**PRUEBAS DE ACEPTACIÓN:**

- El campo de monto debe tener valor, en caso contrario se informará con un mensaje de error.
- La fecha debe ser válida, es decir anterior al momento de la subida, en caso contrario se informará con un mensaje de error.

- **RF5: CLASIFICACIÓN DE GASTOS:** Como product owner quiero permitir a los usuarios clasificar sus gastos en diferentes categorías predefinidas (por ejemplo, alimentación, transporte, facturas, etc.) para facilitar la organización de los mismos. Esto se puede conseguir mediante una selección desde un menú desplegable o mediante etiquetas personalizadas.

**PRUEBAS DE ACEPTACIÓN:**

- En caso de no seleccionar una categoría, se guardará como misceláneo/otros.
- Al seleccionar una categoría, se pide un listado de gastos de esa categoría y aparece el nuevo gasto almacenado.

- **RF6: CATEGORÍAS PERSONALIZADAS:** Como usuario quiero crear categorías personalizadas para adaptarse a sus necesidades específicas de seguimiento de gastos. Esto les dará más flexibilidad para organizar y analizar sus datos financieros.

**PRUEBAS DE ACEPTACIÓN:**

- En caso de intentar crear una nueva categoría que ya está en la base de datos, se informará con un mensaje de error.
- Al crear una nueva categoría, esta aparece en el listado de categorías.

- **RF7: VISUALIZACIÓN DE DATOS:** Como scrum master quiero ofrecer herramientas de visualización de datos, como gráficos y tablas, para mostrar información resumida sobre los gastos del usuario. Esto puede incluir gráficos de barras, circulares o de líneas para representar la distribución de los mismos, tendencias a lo largo del tiempo, etc.

**PRUEBAS DE ACEPTACIÓN:**

- Si el usuario selecciona que quiere ver tendencias en sus gastos pero no ha subido gastos un mínimo de 3 meses se informará de que aún no es posible mostrarla.
- Si el usuario selecciona un tipo de tabla o gráfico sin seleccionar categorías se le mostrará dicho gráfico o tabla incluyendo todas las categorías de su base de datos.

- Si el usuario selecciona 2 o más categorías se le mostrará dicho gráfico solo con esas categorías.

- **RF8: FUNCIONALIDAD DE BÚSQUEDA:** Como usuario quiero buscar algún gasto concreto para facilitar la revisión y el análisis de los datos del individuo.

**PRUEBAS DE ACEPTACIÓN:**

- En caso de no encontrar ninguna coincidencia, no se mostrarán resultados
- En caso de encontrar gastos coincidentes con la búsqueda, se mostrará un listado con las opciones.

- **RF9: FUNCIONALIDAD DE FILTRADO:** Como desarrollador quiero permitir a los usuarios filtrar por categorías teniendo en cuenta la fecha, la cantidad, etc., para hacer más amena la búsqueda.

**PRUEBAS DE ACEPTACIÓN:**

- En caso de introducir una fecha no válida, es decir, posterior a la fecha de la búsqueda, se informará con un mensaje de error.
- En caso de no elegir dos límites para la cantidad, se pide seleccionar si el valor seleccionado es un mínimo o un máximo.

- **RF10: FUNCIONALIDAD DE EXPORTACIÓN DE DATOS:** Como product owner quiero permitir a los usuarios exportar sus datos de gastos (entre ciertas fechas, por categorías, etc) en distintos formatos , como en un PDF, para que puedan realizar análisis adicionales o mantener registros fuera de la plataforma.

**PRUEBAS DE ACEPTACIÓN:**

- En caso de que el usuario intente exportar los datos pero no haya ningún gasto almacenado se recibe un mensaje de error.
- En caso de que el usuario seleccione exportar más de 100 gastos se recibirá un mensaje de error.
- En caso de seleccionar una fecha de inicio y final válidas en las que no haya hecho más de 100 gastos en las categorías que desea exportar se le exportará el pdf correctamente.

- **RF11: CREACIÓN DE OBJETIVOS:** Como diseñador quiero permitir a los usuarios establecer metas de gastos mensuales o presupuestos por categoría para que reciban alertas cuando estén cerca de alcanzar esos límites.

## PRUEBAS DE ACEPTACIÓN:

- En caso de realizar un objetivo y no llegar a completarlo, se notificará al usuario y se recomendarán crear nuevas metas para los siguientes meses.
- Si el usuario sobrepasa su objetivo mensual, se le avisará por si necesita editar la cantidad de gastos máximos de ese mes.

- **RF12: MUESTRA DE PROGRESO:** Como analista quiero que se muestre un resumen para avanzar en un objetivo en concreto o comparar cómo está evolucionando en comparación con el resto de sus objetivos.

## PRUEBAS DE ACEPTACIÓN:

- Será necesaria una mínima cantidad de gastos para poder realizar un resumen.
- El análisis de la evolución de gastos se realizará una vez a la semana.

- **RF13: RECORDATORIOS:** Como scrum master quiero permitir a los usuarios configurar recordatorios para pagos pendientes o fechas límite importantes.

## PRUEBAS DE ACEPTACIÓN:

- Se mandará una notificación al usuario una vez al día durante los últimos 3 días informando sobre el recordatorio que previamente ha configurado.
- El recordatorio se eliminará una vez pase la fecha seleccionada.

- **RF14: NOTIFICACIONES:** Como desarrollador quiero que se pueda permitir el envío de notificaciones por correo electrónico o mensajes dentro de la plataforma para alertar sobre gastos significativos o cambios en el estado financiero para que el usuario habilite si quiere que le lleguen notificaciones en el dispositivo donde lo use.

## PRUEBAS DE ACEPTACIÓN:

- En caso de detectar una evolución contundente se informará al usuario con un mensaje especial.
- En caso de desactivar las notificaciones, solo se notifican los mensajes de máxima importancia.
- Solo se enviarán las notificaciones al correo en caso de tener activada dicha opción.

- **RF15: SOPORTE MULTIMONEDA:** Como analista quiero que para los usuarios que se encuentran en diferentes regiones o países, se proporcione soporte multimonedas, para que el usuario elija la moneda con la cual quiere operar.

**PRUEBAS DE ACEPTACIÓN:**

- Seleccionar una región o país diferente al país de origen y verificar que la opción para elegir la moneda esté disponible en la interfaz de usuario.
- Seleccionar una moneda diferente a la predeterminada y verificar que las cantidades especificadas de los gastos se muestren correctamente en la nueva moneda que has seleccionado.
- Introducir una moneda no compatible y verificar que el sistema muestre un mensaje de error indicando que la moneda no es válida para operar.

- **RF16: BILINGÜE:** Como desarrollador quiero que haya una dualidad de idiomas entre los que elegir: español e inglés, para que el usuario pueda elegir con qué idioma se siente más cómodo usando nuestros servicios.

**PRUEBAS DE ACEPTACIÓN:**

- Verificar que al acceder a la página web, la opción para seleccionar el idioma esté claramente visible y accesible para el usuario.
- Seleccionar uno de los idiomas disponibles y verificar que toda la interfaz de usuario, incluidos los menús, botones, etiquetas y mensajes, se muestre en el idioma seleccionado de manera coherente y correcta.
- Verificar que los formularios de entrada de datos, como los campos de registro de usuario o de ingreso de transacciones, acepten y validen correctamente los caracteres especiales y las letras específicas de cada idioma.

- **RF17: ASISTENCIA Y SOPORTE AL CLIENTE:** Como scrum master quiero establecer canales de comunicación para que los usuarios puedan obtener ayuda, hacer preguntas o informar problemas relacionados con la plataforma. Esto puede incluir un centro de ayuda, correo electrónico de soporte...

**PRUEBAS DE ACEPTACIÓN:**

- Verificar que haya un centro de ayuda claramente visible y accesible desde la página principal de la plataforma.
- Acceder al centro de ayuda y verificar que contenga una variedad de recursos, como preguntas frecuentes, guías de usuario, etc. Organizados de manera clara y fácil de entender.

- **RF18: ACTUALIZACIONES Y MEJORAS CONTINUAS:** Como scrum master quiero comprometerme a mantener la plataforma actualizada con nuevas características, mejoras de rendimiento y correcciones de errores en respuesta a los comentarios de los usuarios y cambios en las necesidades del mercado, para que los usuarios puedan contar con la mejor experiencia con nuestro producto.

**PRUEBAS DE ACEPTACIÓN:**

- En caso de realizar una pausa por mantenimiento en la página web, se notificará a los usuarios con tiempo de antelación para que tengan tiempo necesario de consultar sus gastos antes de iniciarla.
- Si el usuario tiene algunas recomendaciones o ideas que puedan mejorar la página web, escucharemos todas las sugerencias e intentaremos aplicarlas en caso de que sea posible.

- **RF19: INTELIGENCIA ARTIFICIAL:** Como usuario me gustaría tener acceso a una inteligencia artificial para tomar decisiones acertadas y más personalizadas.

**PRUEBAS DE ACEPTACIÓN:**

- Se actualizará dicha ayuda siempre que sea posible.
- Siempre se proporcionará la correspondiente fuente a la hora de dar datos y estudios.
- El usuario tendrá la opción de informar sobre posibles datos erróneos o indicaciones inapropiadas.

- **RF20: VISUALIZACIÓN DE PERFILES:** Como usuario me gustaría visualizar otros perfiles y sus gastos para comparar y ver que puedo mejorar.

**PRUEBAS DE ACEPTACIÓN:**

- Solo serán visibles aquellos perfiles que elijan ser públicos.
- Verificar que se muestran campos como nombre de usuario y correo electrónico.

- **RF21: VISUALIZACIÓN DE CUENTAS Y TARJETAS:** Como usuario me gustaría poder vincular mis cuentas bancarias/tarjetas para tener una gestión más actualizada y automática de los gastos.

**PRUEBAS DE ACEPTACIÓN:**

- Verificar que exista una opción clara y accesible en la interfaz de usuario para vincular cuentas bancarias y tarjetas de crédito/débito.
- Realizar pruebas de seguridad para garantizar que la información financiera del usuario esté protegida adecuadamente y que no haya riesgos de acceso no autorizado a través de las cuentas vinculadas.

- **RF22: CATALOGAR POR GRUPOS:** Como usuario me gustaría tener la opción de catalogar mis gastos como grupales/conjuntos, para poder ver tanto la suma total del gasto, como los participantes en el mismo.

**PRUEBAS DE ACEPTACIÓN:**

- En caso de que un usuario quiera mover o eliminar un gasto de un conjunto, tendría la opción de realizarlo y se le pedirá confirmación en caso de que haya sido por error.
- Verificar que no se permita crear grupos con nombres o elementos duplicados.
- Probar la funcionalidad de catalogar por grupos en diferentes navegadores web y dispositivos para asegurarse de que la experiencia del usuario sea consistente en todas las plataformas.

- **RF23: HISTORIAL DE GASTOS:** Como usuario querría tener acceso a todos mis gastos pasados con análisis de patrones/tendencias para poder ver mi progreso en el tiempo y tomar mejores decisiones a futuro.

**PRUEBAS DE ACEPTACIÓN:**

- Verificar que exista una sección claramente identificada en la interfaz de usuario para acceder al historial de gastos.
- Realizar una búsqueda de transacciones por categoría o período de tiempo específico y verificar que los resultados se filtren correctamente según los criterios seleccionados.
- Verificar que los usuarios puedan personalizar la visualización del historial de gastos según sus preferencias individuales, como la agrupación por categoría o la ordenación por fecha.

- **RF24: APARTADO “SOBRE MÍ”:** Como desarrolladores queremos tener un apartado con información relacionada a nosotros para poder crecer nuestras redes sociales y por tanto nuestro alcance.

**PRUEBAS DE ACEPTACIÓN:**

- Asegurarse de que en dicho apartado aparecen y funcionan correctamente los enlaces a las redes sociales de WiseWallet.
- En caso de que el usuario seleccione ver información sobre nosotros, que aparezca una breve descripción de nosotros junto a los nombres y redes de todos los integrantes del grupo.

- **RF25: CONTENIDO MULTIMEDIA/AUDIOVISUAL:** Como usuario querría tener acceso a un apartado en el que pueda visualizar diferente contenido que me pueda ayudar a entender mejor ciertos conceptos o aprender nuevas estrategias de ahorro y administración del dinero como pueden ser libros, vídeos, artículos, etc.

**PRUEBAS DE ACEPTACIÓN:**

- Verificar que exista un apartado claramente identificado en la interfaz de usuario para acceder al contenido multimedia/audiovisual.
- Probar la funcionalidad de búsqueda dentro del apartado de contenido multimedia y verificar que los usuarios puedan encontrar recursos específicos de manera rápida y eficiente.
- Verificar que el contenido multimedia/audiovisual esté organizado de manera clara y estructurada, con categorías o etiquetas que faciliten la navegación.

- **RF26: SUSCRIPCIONES DE PAGO:** Como product owner me gustaría incluir varios tipos de suscripciones de pago con las cuales el usuario puede desbloquear ciertas funcionalidades extras, además de poseer ventajas respecto a los que no las posean (atención personal 24h, contenido adicional, etc.).

**PRUEBAS DE ACEPTACIÓN:**

- Verificar que una vez obtenida la suscripción, todas las funcionalidades estén disponibles.
- Verificar que se pueda realizar la cancelación de la adquisición semanal/mensual/anual de la suscripción con facilidad .

- **RF27: VISUALIZACIÓN DE MERCADO:** Como product owner me gustaría que se pudiera consultar en una pequeña interfaz tanto los valores de acciones de las diferentes empresas existentes como los de las criptomonedas más populares y con mayores inversiones.

**PRUEBAS DE ACEPTACIÓN:**

- Verificar que los datos visualizados sean de fuentes fiables y actualizadas para que no hayan confusiones o equivocaciones a la hora de realizar una inversión.
- Asegurarse de que se ofrece una comparación gráfica para visualizar las fluctuaciones del mercado a lo largo de semanas/meses/años.

- **RF28: NOTICIAS FINANCIERAS:** Como analista quiero incluir un segmento en el que se encuentre un feed de noticias financieras de los periódicos más prominentes en el tema para que nuestros usuarios se puedan mantener informados sobre las noticias económicas más relevantes del momento.

## PRUEBAS DE ACEPTACIÓN:

- En caso de encontrar una noticia falsa o manipulada, el usuario podrá informar a los desarrolladores.
- El usuario tendrá la opción de seleccionar las editoriales que más le interesen.
- Las noticias solo se mantendrán durante una semana.
- Los usuarios tendrán la opción de descargar las noticias que consideren más interesantes.

- **RF29: FORO DE USUARIOS:** Como scrum master quiero poner un foro de usuarios puedan intercambiar consejos, experiencias o funcionalidades que les ha ayudado, para que de esta manera se puedan comunicar entre ellos mejorando la experiencia como comunidad.

## PRUEBAS DE ACEPTACIÓN:

- Probar la funcionalidad de registro de usuarios en el foro y verificar que sea sencillo y fácil de completar.
- Verificar que los usuarios puedan iniciar nuevas conversaciones (hilos) en el foro sobre diferentes temas relacionados con la gestión financiera personal.
- Verificar que los usuarios puedan responder a los hilos existentes y participar en las conversaciones de manera activa.
- Probar la funcionalidad de moderación del foro, asegurando que los administradores puedan eliminar contenido inapropiado o spam, y tomar medidas contra usuarios que violen las reglas del foro.

- **RF30: RESPALDO AUTOMÁTICO DE DATOS:** Como analista quiero que se realicen copias de seguridad automáticas de los datos del usuario para evitar la pérdida de información en caso de fallos técnicos.

## PRUEBAS DE ACEPTACIÓN:

- Habilitar la opción de respaldo automático y establecer la frecuencia de respaldo.
- Acceder al archivo de copia de seguridad generado automáticamente.

- **RNF1: DESCARGAS SIN CONEXIÓN:** Como usuario me gustaría poder guardar algunos apartados de la página web en el navegador para poder visitarlos sin conexión a internet.

## PRUEBAS DE ACEPTACIÓN:

- Probar la funcionalidad de descarga de contenido y verificar que sea fácil de usar y comprender para los usuarios.
- Verificar que los usuarios puedan acceder al contenido descargado sin conexión a internet y que se muestre de manera correcta y completa.



- **RNF2: DISEÑO RESPONSIVO:** Desarrollar una interfaz de usuario adaptable que funcione bien en diferentes dispositivos, navegadores y tamaños de pantalla, como computadoras de escritorio, tabletas y teléfonos móviles.

**PRUEBAS DE ACEPTACIÓN:**

- Verificar que la interfaz de usuario se adapte de manera adecuada y fluida a diferentes tamaños de pantalla, como computadoras de escritorio, tabletas y teléfonos móviles.
- Verificar que los elementos de la interfaz como botones, menús y formularios, se puedan redimensionar y reorganizar de manera inteligente para adaptarse al tamaño de pantalla sin sacrificar la funcionalidad.

- **RNF3: INTUITIVIDAD:** La página web debe ser fácil de usar y comprender para una amplia variedad de usuarios, independientemente de su nivel de experiencia técnica. Esto implica un diseño intuitivo de la interfaz de usuario, navegación clara y consistente, y etiquetas descriptivas.

**PRUEBAS DE ACEPTACIÓN:**

- Verificar que personas con un rango relativamente grande pueda utilizar correctamente la mayoría de las funcionalidades desde el primer momento como registrar la cuenta bancaria, personalizar su cuenta, etc.
- Evitar el uso excesivo de interfaces llenas de conceptos no muy comunes entre la mayoría de los usuarios y de desplegables enormes que tiendan a confusiones.

- **RNF4: SEGURIDAD Y PRIVACIDAD:** Implementar medidas de seguridad robustas para proteger los datos de los usuarios, como el cifrado de datos, la prevención de ataques de seguridad, la autenticación de dos factores, etc. Además, asegurarse de cumplir con las regulaciones de privacidad de datos aplicables.

**PRUEBAS DE ACEPTACIÓN:**

- Se mantendrán actualizadas las medidas de seguridad en la medida de lo posible.
- Cada cierto tiempo se recordará al usuario sobre la importancia de activar los dos factores.

- **RNF5: MANTENIBILIDAD:** Facilitar la capacidad de mantenimiento y actualización del sistema a lo largo del tiempo, mediante una arquitectura modular, código limpio y documentación detallada.

**PRUEBAS DE ACEPTACIÓN:**

- Probar la facilidad de agregar nuevas funcionalidades o realizar cambios en el sistema sin requerir modificaciones extensas en el código existente.
- Verificar que el código fuente esté bien estructurado y siga buenas prácticas de

programación, como la coherencia en la nomenclatura, la modularidad y la legibilidad del código.

- Probar la facilidad de realizar pruebas de unidad, integración y funcionales en el sistema, utilizando frameworks de pruebas adecuados y herramientas de automatización.

- **RNF6: FUNCIONAMIENTO EFICIENTE:** Como desarrollador quiero proporcionar el mínimo tiempo de espera posible para garantizar una experiencia fluida a los usuarios.

#### PRUEBAS DE ACEPTACIÓN:

- Medir y verificar los tiempos de respuesta del sistema para diferentes acciones y operaciones, asegurándose de que estén dentro de los límites aceptables.
- Realizar pruebas de carga para evaluar cómo el sistema maneja un aumento en el número de usuarios concurrentes y asegurarse de que el rendimiento no se degrade significativamente.
- Realizar pruebas de rendimiento en diferentes navegadores y dispositivos para asegurarse de que la velocidad de carga de la página sea consistente y rápida en todas las plataformas.

<<RF>> Inicio de Sesión	<<RF>> Registro de usuario	<<RF>> Gestión de usuarios	<<RF>> Subida de gastos	<<RF>> Clasificación de gastos	<<RF>> Categorías personalizadas
ID = "RF1"	ID = "RF2"	ID = "RF3"	ID = "RF4"	ID = "RF5"	ID = "RF6"
<<RF>> Visualización de datos	<<RF>> Funcionalidad de búsqueda	<<RF>> Funcionalidad de filtrado	<<RF>> Funcionalidad de exportación de datos	<<RF>> Creación de objetivos	<<RF>> Muestra de progreso
ID = "RF7"	ID = "RF8"	ID = "RF9"	ID = "RF10"	ID = "RF11"	ID = "RF12"
<<RF>> Recordatorios	<<RF>> Notificaciones	<<RF>> Soporte multimoneda	<<RF>> Bilingüe	<<RF>> Asistencia y soporte al cliente	<<RF>> Actualizaciones y mejoras continuas
ID = "RF13"	ID = "RF14"	ID = "RF15"	ID = "RF16"	ID = "RF17"	ID = "RF18"
<<RF>> Inteligencia artificial	<<RF>> Visualización de perfiles	<<RF>> Visualización de cuentas y tarjetas	<<RF>> Catalogar por grupos	<<RF>> Historial de gastos	<<RF>> Apartado "sobre mí"
ID = "RF19"	ID = "RF20"	ID = "RF21"	ID = "RF22"	ID = "RF23"	ID = "RF24"
<<RF>> Contenido multimedia	<<RF>> Suscripciones de pago	<<RF>> Visualización de mercado	<<RF>> Noticias financieras	<<RF>> Foro de usuario	<<RF>> Respaldo automático de datos
ID = "RF25"	ID = "RF26"	ID = "RF27"	ID = "RF28"	ID = "RF29"	ID = "RF30"
<<RNF>> Descargas sin conexión	<<RNF>> Diseño responsivo	<<RNF>> Intuitividad	<<RNF>> Seguridad y privacidad	<<RNF>> Mantenibilidad	<<RNF>> Funcionamiento eficiente
ID = "RNF1"	ID = "RNF2"	ID = "RNF3"	ID = "RNF4"	ID = "RNF5"	ID = "RNF6"

***SECCIÓN 6: Herramientas software usadas durante la realización del proyecto. Indicar las herramientas software que se han usado para desarrollar el proyecto hasta la fecha. Cualquier tipo de herramienta: de comunicación, trabajo colaborativo, elaboración de documentos, etc.***

Las herramientas que hemos utilizado hasta el momento son: Whatsapp, Trello, Visual Paradigm, Drive, Gmail y GitHub.