Memoria

**1.-Introducción**

Inicialmente, nos pusimos a debatir sobre qué herramientas serían más convenientes para el *Product Backlog* y el *Sprint Backlog*.

Tras haber acordado que usaríamos *Jira* para gestionar el *Product Backlog* y el *Sprint Backlog*, el equipo se dividió en dos grupos. Uno de ellos estaba formado por el ***Product Owner* y el *Scrum Master***, que se dedicaron a escribir sobre las historias de usuario que formarían posteriormente el *Product Backlog*.

Al principio, quedaron bastantes Historias de Usuario épicas, pero con ayuda de todo el grupo se fueron simplificando y dividiendo en unas más simples, que serían sobre las que se harían los criterios de aceptación posteriormente.

Tras ser validadas por todo el equipo, serían acompañadas de los criterios de aceptación.

El otro grupo, que estaba formado por los **desarrolladores**, se dedicó a practicar con las herramientas y tecnologías que se usarían en el proyecto (mientras el Product Owner y el Scrum Master elaboraban las Historias de Usuario) para más tarde crear el prototipo.

Se acordó que el prototipo tendría varias funciones básicas que permitiera tanto manejar la parte de interfaces gráficas de Netbeans, como crear y mantener una base de datos con SQLite (previamente creada) en lenguaje Java. Además, habría que hacer un *login* al iniciar el prototipo.

Tras probar el prototipo, analizar y corregir errores, se preparó para que se pudiera ejecutar desde un *.jar* y generara automáticamente la base de datos.También se elaboraron las instrucciones de ejecución correspondientes, y se redactó las dificultades encontradas a la hora de realizar el prototipo.

**2.-Historias de usuario**

En los siguientes apartados explicaremos las herramientas usadas para que el Product Owner y el equipo de desarrollo puedan gestionar el Product Backlog y el Sprint Backlog, respectivamente.

También se explicará el proceso de validación de las Historias de Usuario, y cómo el Product Owner las priorizó posteriormente.

Como añadido, hemos incluido los Mockups de la aplicación, y perfiles de usuario.

**a)Herramienta usada para gestionar el Product Backlog y el Sprint Backlog**

Para el *Sprint Backlog* y *Product Backlog* decidimos usar **Jira**.

Una de nuestras opciones era usar *Trello* para la gestión del Sprint Backlog, debido a que su uso era sencillo. Sin embargo, Jira tiene más funcionalidades, tiene un uso bastante extendido, y con otras herramientas como GitHub (que también usaremos, pero para el control de versiones), es más difícil dividir el *Product Backlog* en tareas.

Primero probamos *Trello*, que resultaba ser muy amigable para el usuario, pero no tenía demasiadas funcionalidades. Posteriormente, comprobamos que en GitHub había que crear las tareas de cada Historia de Usuario a mano, por lo que vimos que era muy costoso.

Por último, usamos *Jira*. Siguiendo algunos tutoriales, no resultaba demasiado difícil de manejar, y era una herramienta bastante útil para metodologías ágiles.

Tras probar las herramientas anteriormente mencionadas y debatir cuál sería la más adecuada, decidimos usar ***Jira***, pues a pesar de que es algo complejo de usar al principio por la cantidad de funcionalidades que ofrece, teníamos la oportunidad de usar una herramienta más completa y a la que se le suele dar uso de forma profesional.

**b) Validación de Historias de Usuario**

**c)** **Priorización de las Historias de Usuario**

**d)** **Mockups creados**

[Insertar mockups, y explicarlos brevemente]

**e) Perfiles de usuario**

[Insertar perfiles de usuario]

**3.-Prototipo**

En este apartado se explicará tanto las tecnologías probadas en el prototipo que realizamos, como su desarrollo, problemas encontrados, cómo se han resuelto, y finalmente se explicará cómo ejecutarlo y probar las funcionalidades que este ofrece.

**a) Herramientas y tecnologías probadas**

Además de las herramientas usadas para que el *Product Owner* gestione el *Product Backlog*, debatimos sobre qué herramientas usar para desarrollar, y para gestionar las versiones del código de proyecto, documentación, etc.

El *Product Owner* indicó que sería una aplicación de escritorio, por lo que para programar barajamos varias opciones como **Eclipse**, o **Netbeans**, ya que son plataformas para programar en Java bastante conocidas y amigables. Elegimos esta última, pues aunque ya habíamos trabajado con Eclipse anteriormente, Netbeans incorporaba el *JavaFx*, una herramienta que sirve para diseñar interfaces gráficas. En caso de elegir Eclipse, hubiéramos tenido que instalar el plugin aparte, por lo que nos decantamos por la elección más fácil a la hora de instalación (**Netbeans**), ya que esta plataforma no difiere mucho de Eclipse.

Para el servidor de la base de datos, estuvimos debatiendo sobre el uso de **Xampp** y **Wampp**, debido a que a la hora de hacer la conexión de la base de datos con el código es bastante simple, y varios integrantes del grupo lo han utilizado previamente, lo que puede a ayudar a resolver dudas en este periodo de prueba de herramientas.

Otra ventaja que aportaban estos programas es que, al unificar varios servidores, permiten elegir en la instalación qué servidor queremos usar.

Ya que el prototipo sería ejecutado desde un *.jar*, y se tendría que generar la base de datos a partir de dicho ejecutable sin depender de software externo, se propuso también el uso de **SQLite**, ya que va integrado dentro del propio proyecto, y sería sencillo realizar la conexión a la base de datos.

Debido a las restricciones de la entrega del prototipo, y habiendo probado las plataformas mencionadas, decidimos usar **SQLite** para las bases de datos. Tras acordar esto, se comentó que con **SQLiteStudio** se podría gestionar fácilmente la base de datos creada.

Comprobamos que era amigable, y tenía una interfaz simple, por lo que decidimos usarlo para la gestión de la base de datos, (además de usar el SQLite).

**b)** **Desarrollo del prototipo**

A la hora de crear el prototipo nos dividimos en pequeños subgrupos dentro de los propios desarrolladores, y nos repartimos equitativamente pequeñas partes de lo que sería el prototipo. Posteriormente integraríamos todo.

**c) Instrucciones de ejecución y cómo probar el prototipo**

Para ejecutar el proyecto, primero es necesario ejecutar el .jar, e iniciar sesión con la contraseña *“admin”*

**4.- Dinámica grupal**

Nuestro grupo desarrolló una dinámica grupal para mejorar la convivencia y trabajar mejor en equipo.

A continuación, haremos una breve descripción de la dinámica, explicaremos cómo se desarrolló, y cuáles fueron los resultados de esta.

**a) Descripción de la dinámica**

**b) Desarrollo**

**c) Resultados**