PRÁCTICA 3

User Story Mapping

The Video Game Box

INTEGRANTES DEL GRUPO:
MARIO CAMPOS SOBRINO
CARLOS CARNERO MÉRIDA
JOSÉ DÍAZ REVIEJO
DAVID ELÍAS PIÑEIRO
CARLOS GÓMEZ LÓPEZ
ÁLVARO GÓMEZ SITTIMA
JULIÁN MOFFATT
JUAN ROMO IRIBARREN
JAVIER DE VICENTE VÁZQUEZ
GONZALO VÍLCHEZ RODRÍGUEZ

0. Índice

1. Creación del User Story Mapping

1.1 Introducción

Para la realización de esta práctica comenzamos reuniéndonos todos los miembros del equipo para plantear cómo haríamos el *User Story Mapping*.

Lo primero que hicimos fue consensuar qué herramienta era la más adecuada para llevar a cabo esta tarea. Tras testear varias opciones, pensamos que *Miro* era la que más se ajustaba a lo que necesitábamos y empleamos varias horas en aprender a utilizarlo y ratificar nuestra decisión, que detallaremos más adelante.

Posteriormente, empezamos a plantear nuestro *User Story Mapping* con una lluvia de ideas en la que, tras agruparlas, decidiríamos sus prioridades dentro del mapeado. A partir de ahí, fuimos desarrollándolo y construyéndolo entre todos de manera progresiva mientras nos dividimos para ir recopilando todos los aspectos que debíamos incluir en la memoria de la práctica.

Para comenzar, Gonzalo se encargó de elaborar una justificación adecuada de la herramienta utilizada, ya que indagó un poco más que el resto en conocer bien las funcionalidades de la misma.

Durante el proceso, David, Mario y Julián se encargaron de ir explicando el criterio empleado para priorizar las distintas tareas. Mientras tanto, al mismo tiempo, Carlos G., Álvaro y José se encargaron de llevar un seguimiento del proceso, detallando y anotando todos los elementos detectados en el proceso. También, Juan se encargó de anotar qué funcionalidades dieron dificultades a la hora de definirlas, y Javier controló el qué se hacía y el tiempo empleado por cada sesión realizada.

Para terminar Carlos C., elaboró la introducción basándose en anotaciones previas sobre cómo fuimos llevando a cabo la práctica y lo que llegó a realizar cada miembro del equipo.

1.2 Mapa de Historias de Usuario

El *Mapa de Historias de Usuario* o *User Story Mapping* es una técnica que permite a los gerentes de producto, desarrolladores y diseñadores obtener un método con el que logren alcanzar las expectativas del usuario de cara al producto y la clara representación de las mismas.

Para ello en este punto se va a explicar todo lo relacionado con el desarrollo del mapeado.

1.2.1 Herramienta empleada

Para llevar a cabo el Mapa de Historias de Usuario hemos usado la herramienta *Miro*. Esta es una herramienta online que permite la creación de

diagramas de forma colaborativa y en tiempo real, con una interfaz gráfica y sencilla en forma de pizarra virtual. En nuestro caso, hemos utilizado la licencia gratuita que ofrecen a estudiantes, que elimina las restricciones de la licencia gratuita a nivel general.

También, hemos decidido utilizar está herramienta principalmente porque permite trabajar en tiempo real con otros miembros del equipo, lo cual facilita trabajar de forma concurrente. Esta aplicación se puede, a su vez, integrar con *Jira* (herramienta que vamos a utilizar para la gestión de tareas del proyecto) para insertar en ésta los diagramas y poder visualizarlos y editarlos sin salir de la aplicación.

1.2.2 Proceso de creación del Mapa de Historias

Con respecto a cómo se ha llevado a cabo la creación del mapa, vamos a detallar la duración del proceso, los elementos que se han llegado a detectar en cada paso y por último las dificultades en algunas de las funciones a la hora de definirlas.

Empezamos comentando el tiempo que hemos necesitado a la hora de pensar y escribir todas las funciones, ordenarlas por prioridades, agruparlas y ya finalmente situarlas en el mapa.

1.2.2.1 Tiempo invertido

El tiempo empleado total ha sido de *ocho horas*. Este tiempo es la suma de tres sesiones de trabajo, en las cuales, la primera duró dos horas y estuvimos todo el equipo reunido aprendiendo a cómo utilizar *Miro* y decidir si era buena opción para realizar en dicha aplicación el mapa. La siguiente reunión duró dos horas y media, las cuales invertimos en desarrollar la base del *User Story Mapping* dentro de *Miro*. Para finalizar, realizamos una última sesión, donde básicamente se completó el *User Story Mapping*, tardando tres horas y media.

1.2.2.2 Elementos detectados

A continuación, explicaremos todos los elementos encontrados a medida que íbamos desarrollando el mapa.

Empezamos pensando en funciones básicas como buscar, ver detalles, iniciar sesión y funciones que pueda gestionar un usuario registrado. De estas funciones básicas, salieron derivadas de ellas, es decir, por ejemplo, en buscar, obtuvimos una función por cada entidad que tenemos (juegos, boxes y usuarios). También, por cada búsqueda, lo más razonable era poder ver los detalles de los datos obtenidos de la búsqueda, por ello incorporamos las funciones de ver detalles de cada entidad, y también se nos ocurrió que se podrían realizar búsquedas en función de filtros, dando así mayor accesibilidad y facilidad a la hora de obtener los resultados. Para dichos resultados, pensamos que sería más cómodo si se pudiesen ordenar los datos obtenidos en función de algunos de sus atributos.

Con respecto al inicio de sesión, nos dimos cuenta de que, para iniciar sesión, primero debes de poder registrarte, por ello salió esa función, junto con los distintos mecanismos con los que se puede iniciar de sesión. Obviamente, pensamos: *si se inicia sesión, también se podrá cerrar*; por lo que otra función que incorporamos a la lista de funciones fue el cerrar sesión.

A continuación, pensamos en qué funcionalidades va a poder llegar a realizar el usuario. Entre ellas, las primeras que salieron fueron el crear y borrar boxes, y el añadir y eliminar juegos de una box. Y también el cómo gestionar el juego en función de un estado que le asocie el usuario.

También, se nos ocurrió el poder poner una box en privada o pública, y poder llegar a interactuar con las boxes y juegos de otros usuarios, siguiendo a las boxes y valorando a los juegos.

Para finalizar, pensamos en algunas funciones que veíamos difíciles de llegar a implementar, pero ya decidiríamos a la hora de priorizar si las llegábamos a desarrollar o no, estas funciones estaban relacionadas con hacer Reviews, recuperar una box que se haya eliminado, añadir juegos que no se encuentre dentro de la base de datos y poder customizar el usuario su perfil.

Una vez terminamos la lluvia de ideas, nos quedaron todas las funciones registradas en la *tabla 1.2.2.2.1*.

Funciones detectadas	
Búsqueda de juegos por nombre	Búsqueda de boxes por nombre
Búsqueda de usuario por nombre	Atributos básicos del juego
Atributos básicos de la box	Atributos básico del usuario
Registrarse con el correo	Iniciar sesión con correo/contraseña
Crear box	Añadir un juego a una box
Cerrar sesión	Ver juegos de la box
Búsqueda de juegos por categoría	Búsqueda de boxes por categoría
Ordenar juegos por número de likes	Ordenar boxes por seguidores
Ordenar juegos por fecha	Búsqueda de juegos por estado
Ver número de likes del juego	Ver número de seguidores de la box
Ver boxes del usuario	Seguir box
Dar like al juego	Borrar box
Borrar juego de una box	Gestionar privacidad de una box
Ver juegos similares de un juego	Búsqueda de juegos por estado
Ordenar juegos por fecha	Búsqueda de juegos por compañía
Iniciar sesión con Google	Asignar al juego un estado
Búsqueda de juegos por plataforma	Iniciar sesión con Facebook
Búsqueda de juegos por intervalo de fechas	Hacer Reviews
Customizar perfil	Recuperar box borrada
Añadir juegos que no están en la base de datos	

Tabla 1.2.2.2.1 Funciones Detectadas

1.2.2.3 Funcionalidades dificultosas

Por último, comentaremos todas las dificultades que tuvimos con algunas funcionalidades mientras trabajamos en el mapa.

La primera de todas y consideramos una de las más importante, ya que no ha afectado solo a una, si no a varias. Esta dificultad has sido el cambiarle el nombre de colección a *box*, la razón por la que se ha cambiado ha sido porque como en un futuro vamos a usar *MongoDB*, íbamos a tener problemas por llamar a nuestras listas de juegos colecciones, ya que en *MongoDB* también llama a colecciones donde se almacenan registros individuales.

Y la otra única dificultad que tuvimos fue la de *Borrar Box*, ya que, a la hora de asignarla una prioridad, hubo debate sobre si tenía que estar en la misma release que *Crear Box* o una más abajo. Se acabó poniendo en la segunda release, porque en sí no es tan importante ni necesaria como para formar parte del *Walking Squeleton*, y el programa puede funcionar sin ningún problema sin dicha función.

1.2.3 Priorización de las tareas

Para finalizar el User Story Mapping, priorizaremos las funciones a través de la técnica *MoSCoW*, que consiste en agrupar las *Historias de Usuario (HU)* en cuatro grupos en función de su prioridad:

- **Must Have:** estas tareas son indispensables para el éxito del proyecto y por tanto deben llevarse a cabo en primer lugar.
- **Should Have:** agrupa las tareas importantes para realizar una vez finalizadas las tareas indispensables para el funcionamiento de la aplicación. Aportan un valor añadido al proyecto y contribuyen al logro de los objetivos, pero no impiden poner el proyecto en marcha si no se desarrollan.
- **Could Have:** funcionalidades que estaría bien tener si hay tiempo una vez finalizadas las tareas de las dos primeras categorías. Su realización no debe afectar a las demás tareas.
- **Won't Have:** *HU* que el cliente ha solicitado inicialmente pero que se descartan por falta de recursos y/o falta de tiempo. Podría considerarse el desarrollo de algunas de estas funcionalidades si cambian las circunstancias, por lo que es importante tenerlas registradas.

Hemos empleado *MoSCoW* como técnica de priorización pues ayuda a todos los involucrados en el proyecto a definir la importancia y las razones por las que el requisito a implementar es importante.

Para priorizar todas las tareas de la aplicación hemos agrupado dichas tareas por funcionalidad, como se podrán observar en el *UserStoryMapTheVideoGameBox.pdf* . Los criterios para priorizar las tareas dentro de cada funcionalidad son los que se van a desarrollar a continuación.

A la hora de *buscar*, hemos considerado que la búsqueda por nombre es la más importante al ser la manera más rápida de acceder a un juego/box específico. En segundo lugar, nuestra aplicación debería también tener la posibilidad de buscar juegos/boxes por categoría y buscar además juegos por compañía desarrolladora. Como funcionalidad adicional, contemplemos añadir la posibilidad de buscar juegos por estado (pendiente, completado o jugando), por plataforma y por fecha de lanzamiento.

A pesar de que *ordenar* forma parte de la búsqueda, al tener varias opciones de ordenación las describimos aparte (destacar que la ordenación es una tarea importante, pero no prioritaria, por eso se sitúa en el *Should Have*). La forma principal de ordenar un juego es según su número de likes y ordenar un box por número de seguidores, ya que es frecuente que un usuario quiera ordenar juegos o boxes en función de su popularidad. Sería deseable también el poder ordenar juegos por fecha de lanzamiento y así dar la posibilidad al usuario de buscar juegos que se hayan creado dentro del intervalo especificado.

Una vez que el usuario ha realizado la búsqueda deseada, es imprescindible poder *ver las características* de cada uno de los elementos buscados. Lo primordial al mostrar los detalles de un juego, box o usuario es el mostrar sus atributos básicos (nombre, fecha de creación, …) y también mostrar los juegos que componen una box. Sería adecuado también poder ver el número de likes y juegos similares de un juego, así como el ver el número de seguidores de una box y el número de boxes creadas por un usuario.

Continuando con los criterios, sabemos que gran parte de la funcionalidad de la aplicación depende de si el usuario está registrado o no. Entonces, a la hora de realizar el *registro*, el correo es la forma más sencilla de realizar dicha acción.

Una vez el usuario se ha registrado, la forma prioritaria de *iniciar sesión* en la aplicación es a través del correo y una contraseña guardados al registrarse. Podremos añadir también la posibilidad de acceder con la cuenta de Google o la de Facebook.

A la hora de interactuar, vamos a diferenciar entre la interacción con la box y con un juego. Empezaremos, con la *interacción con una box*, la cual una forma de interaccionar con ella es el tener la opción de seguirla, ya que facilitará el acceso a ella en futura ocasiones. Es una tarea importante pero no imprescindible, por ello se encuentra en *Should Have*.

Con respecto a la *interacción con un juego*, se sigue el mismo criterio que la anterior funcionalidad, ya que añadiremos la opción de dar like a un juego específico, pero no es función primordial.

Una vez un usuario se encuentra dentro de la aplicación, es imprescindible el poder *gestionar boxes*, es decir el crear boxes y añadir juegos a una box. También es importante que el usuario pueda borrar sus boxes creadas, eliminar juegos de dichas boxes y gestionar la privacidad de cada una de ellas.

No solo se pueden gestionar boxes, si no que también se pueden *tratar los juegos*, por ello consideramos la posibilidad de asignar al juego un estado (pendiente, completado o jugando) de manera que el usuario tenga más opciones a la hora de buscar juegos. Esta acción, pese a facilitar el uso de la aplicación, es completamente prescindible.

Para finalizar *cerrar sesión*, se considera imprescindible en la aplicación.

Las funciones que se han asignado como imposibles de implementar por falta de recursos o tiempo son hacer Reviews, customizar perfil, recuperar una box ya borrada y añadir juegos que no se encuentran en la base de datos. Todas estas funcionalidades no se pueden implementar por la misma limitación, que es la base de datos, ya que tiene una capacidad máxima, y si se llegaran a implementar algunas de estas funciones podrían llegar a ocuparla al completo, produciendo así un gran problema.

El resultado de la agrupación de la tareas según su funcionalidad queda reflejado en la *tabla 1.2.3.1*.

Must Have	Should Have	Could Have	Won't Have
Búsqueda de juegos por nombre	Búsqueda de juegos por categoría	Búsqueda de juegos por estado	Hacer Reviews
Búsqueda de boxes por nombre	Búsqueda de boxes por seguidores	Búsqueda de juegos por plataforma	Customizar perfil
Búsqueda de usuarios por nombre	Búsqueda de boxes por categoría	Búsqueda de juegos por intervalo de fechas	Recuperar box borrada
Ver atributos básicos del juego	Ordenar juegos por número de likes	Ordenar juegos por fecha	Añadir juegos que no están en la base de datos
Ver atributos básicos de la box	Ver número de likes de un juego	Iniciar sesión con Google	
Ver juegos de la box	Ver juegos similares	Iniciar sesión con Facebook	
Ver atributos básicos del usuario	Ver número de seguidores de una box	Asignar al juego un estado	
Registrarse con el correo	Ver boxes de un usuario		
Iniciar sesión con correo/contraseña	Seguir una box		
Crear una box	Borrar una box		
Añadir un juego a una box	Borrar juego de una box		
Cerrar sesión	Gestionar privacidad de una box		

Tabla 1.2.3.1 Permisos MoSCoW