

PRÁCTICA 7

Primer Sprint



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. Introducción.....	2
2. Product Backlog	3
2.1. Actualizaciones del Product Backlog	3
2.2. Explicación de prioridades primera release	4
3. Reunión de planificación del sprint	5
3.1. Sistema de votación.....	6
3.2. Descomposición de las HU en tareas	8
4. Sprint Backlog	9
5. Desarrollo del Sprint.....	23
6. Reuniones diarias.....	28
6.1. Daily scrum 17-03-2022.....	29
6.2. Daily scrum 20-03-2022.....	29
6.3. Daily scrum 22-03-2022.....	29
6.4. Daily scrum 24-03-2022.....	29
6.5. Daily scrum 27-03-2022.....	29
6.6. Daily scrum 28-03-2022.....	30
7. Backlog con Impedimentos	30
7.1. Un miembro del equipo se va de viaje	30
7.2. Lesión de un miembro del equipo	30
7.3. Dificultad para resolver algunas dudas con el Product Owner.....	30
7.4. Falta de formación	30
7.5. La red de la facultad bloquea el acceso a MONGO DB	30
7.6. Distintos niveles de conocimiento técnico por parte del Equipo de Desarrollo	31
8. Burndown Chart	31
9. Reunión de revisión	32
10. Reunión de retrospectiva.....	34
10.1. Recopilación de información del Sprint	34
10.1.1. Cosas bien realizadas.....	34
10.1.2. Cosas mal realizadas.....	34
10.2. Objetivos para el próximo Sprint.....	34
10.3. Plan de mejora	35
10.4. Conclusión de la reunión.....	35

1. Introducción

El trabajo realizado en esta práctica ha consistido en el desarrollo del primer *Sprint* de los dos que vamos a realizar. El *Sprint* ha consistido en dos semanas y dos días, siguiendo el ciclo de desarrollo de *Scrum*, con el objetivo de obtener un producto funcional y testeado.

El orden en el que procedimos comenzó con la *reunión de planificación del Sprint* el día 15 de marzo, decidiendo el *Sprint Goal* y el *Sprint Backlog*. Una vez establecidos estos criterios, empezamos con el desarrollo del *Sprint*, en el que cada miembro del *Equipo de Desarrollo* escoge una tarea por orden de prioridad, comenzando a trabajar en ella. Una vez terminada pasa a la siguiente tarea libre por prioridad. El siguiente paso es la reunión de revisión del *Sprint*, realizada el día 29 de marzo, presentando todo lo que se ha realizado durante el *Sprint*. Finalmente se produce la reunión de retrospectiva del *Sprint*, también el día 29 de marzo, enfocándose más en el proceso que en el producto.

Por otra parte, cabe destacar algunas de las dificultades encontradas y la manera en las que se han solventado. Brevemente, un gran impedimento fue el hecho de que un miembro del equipo se fue de viaje y tuvimos que cubrir su parte, por lo que durante algunos días teníamos una persona menos de trabajo. Por ello tuvimos que reducir un poco el *Sprint Goal*. Otro problema fue la dificultad dada en algunas situaciones a la hora de comunicarse con el *Product Owner (PO)*. Para mitigar esto, tratamos de colaborar apoyando al *PO* con aquellas dudas que tuviese.

También nos hemos topado con una serie de problemas técnicos, como pueden ser la restricción por parte de la red de la Facultad en cuanto al acceso a nuestro servidor donde está alojada la base de datos o la falta de conocimientos en algunos aspectos. Afortunadamente, pudimos poner fin a estas inconveniencias con el uso de una *VPN* y también con explicaciones técnicas a quién tuviese dudas o problemas.

Finalmente, la memoria se ha estructurado conforme a las indicaciones dadas por la memoria de la práctica. Lo primero que se va a detallar va a ser el *Product Backlog*, actualizado a la nueva versión realizada para este *Sprint* y teniendo en cuenta las correcciones establecidas en la anterior práctica.

Después se detallará la reunión de planificación del *Sprint*, describiendo como se produjo y a qué conclusiones llegamos a la hora de estimar. En esta reunión queda plasmado el *Sprint Backlog*, que es lo siguiente que se detallará en la memoria. En este, se explica la división en tareas de las distintas *Historias de Usuario (HU)*. Lo siguiente a destacar es el *desarrollo del Sprint*, apartado en el cual se explica cómo se produjo, además de qué tareas e *HU* se consiguieron completar. También se detallan las reuniones diarias, es decir, las *Daily Scrum*, detallando su procedimiento y contenido.

Por otra parte, se incluye el *Impediment Backlog* listando las situaciones que impidieron la progresión del proyecto. Asimismo, el *Burndown Chart* estará

explicado y detallado. Y por último, *las reuniones de revisión y de retrospectiva* estarán aclaradas, mostrando la duración de estas mismas además de los participantes y otros detalles.

2. Product Backlog

El *Product Backlog* es un documento vivo y cambiante que hemos adaptado con respecto a la última versión según las indicaciones de la corrección y detalles que ha decidido añadir o reconsiderar el *Product Owner (PO)* durante el desarrollo del *Sprint*.

El documento con toda la información del *Product Backlog* se encuentra en la carpeta de entrega, ya que ocupaba bastante tamaño.

2.1. Actualizaciones del Product Backlog

Decidimos sustituir todas las referencias en el proyecto del término *Categoría* por *Género*, considerando que en el ámbito de los videojuegos es más acertado utilizar el segundo. Hemos actualizado descripciones, atributos en criterios de aceptación y nombres de *HU* como resultado del cambio.

Hemos sustituido el término *Categoría* por *Género* en los atributos de los criterios de aceptación en las *Historias de Usuario* de *Ver atributos básicos del juego*, *Ver atributos básicos de la box*, *Crear box* y *Modificar box*.

Los nombre de las *HU* de *Búsqueda de juegos por categoría* y *Búsqueda de boxes por categoría* han pasado a ser *Búsqueda de juegos por género* y *Búsqueda de boxes por género* respectivamente por este criterio.

A continuación, se detallaron las *HU* donde no estaba correctamente especificado como se presentaban los datos; se añadieron detalles que no fueron barajados con anterioridad y se plasmaron reconsideraciones que se produjeron durante el *Sprint*. Los cambios realizados se detallan a continuación.

Aunque el *Product Owner* consideró aceptable el rendimiento en las pruebas de aceptación de la *HU* de *Búsqueda de juegos por nombre*, se acordó establecer en los criterios de aceptación el mínimo de caracteres a introducir en cuatro para optimizar los tiempos de búsqueda. Esto consigue un mejor filtrado de resultados optimizando los tiempos de ejecución al listar los juegos.

En la *Historia de Usuario* de *Crear box* se añadió la posibilidad de que un usuario seleccione múltiples géneros para una box. Asimismo, se ha añadido el atributo de elegir privacidad eliminando la *HU Gestionar privacidad*, la cual pretendía implementar esto en el futuro. Además, se han quitado las portadas personalizadas de las boxes para no complicar la implementación y el almacenamiento de datos.

Por esto, en la *HU* de *Modificar box* se ha suprimido la posibilidad de modificar la portada de una box ya que estas no van a ser modificables.

Se ha cambiado el orden de visualización de los datos en las *Historias de Usuario* de [Ver juegos de la box](#) y [Ver boxes del usuario](#). En la primera, los juegos se mostrarán en orden inverso en que fueron añadidos a la box (se mostrarán primero los añadidos más recientemente). En la segunda, las boxes de un usuario se mostrarán por orden inverso de creación (se mostrarán primero las boxes más actuales).

Este mismo criterio se utilizará en las HU de [Búsqueda de boxes por nombre](#) y [Búsqueda de boxes por género](#), donde las boxes se mostrarán también por orden inverso de creación.

Además, en la HU de [Búsqueda de boxes por género](#), se mostrarán boxes que tengan el género buscado entre uno de sus múltiples géneros.

En la *Historia de Usuario* de [Ver número de seguidores de la box](#), en vez de ser una característica que consultar aparte, el número de seguidores de una box se mostrará junto con los demás atributos de la box. Esto mismo ocurre con la HU de [Ver número de likes de un juego](#), de forma análoga con el número de likes.

Por último, en las *Historias de Usuario* para la búsqueda de Juegos ([Búsqueda de juegos por género](#), [Búsqueda de juegos por compañía](#), [Búsqueda de juegos por estado](#), [Búsqueda de juegos por intervalo de fecha](#), [Búsqueda de juegos por plataforma](#)) y en la de [Ver juegos similares a un juego](#), los criterios de aceptación se han modificado para establecer que los juegos se mostrarán en el orden en el que fueron añadidos a la base de datos (*default*).

2.2. Explicación de prioridades primera release

En referencia a la parte de la corrección de la [práctica 6](#): “Habéis cambiado algunas prioridades, aunque sigue sin estar claro algunas, como, por ejemplo, que sea más prioritario buscar un juego, que añadirlo a una box”, explicaremos los criterios del *Product Owner* para priorizar las HU.

Para la primera release, las HU más prioritarias son [Búsqueda de juegos por nombre](#) y [Ver atributos básicos del juego](#). Esto se debe a que no requieren registro previo para estar disponibles y permiten un primer acercamiento a la aplicación a los nuevos usuarios. Si estos quieren expandir la experiencia deberán registrarse.

A continuación, consideramos prioritarias las HU relacionadas con el funcionamiento básico de las boxes. A diferencia de otros servicios en los que se puede simplemente consultar detalles de los juegos existentes, queremos que la identidad de nuestra aplicación sea la capacidad de clasificar juegos mediante boxes y las posibilidades que esto ofrece. Por tanto, queremos empezar a implementarlas pronto en la aplicación para acoplarlas correctamente. El usuario podrá crear boxes, añadir juegos y ver lo que contienen en cualquier momento con la implementación de las HU [Crear box](#), [Añadir juego a la box](#) y [Ver juegos de la box](#).

Con la HU *Búsqueda de boxes por nombre* introducimos la primera herramienta para acceder a las boxes creadas, ya que aún no están ligadas a quien las creó. Esto será así hasta que se implemente el bloque de HU de logueo que permite tener una cuenta de usuario.

Por último, nos parece adecuado que los usuarios se puedan crear una cuenta, iniciar sesión y cerrar sesión mediante las HU *Sistema de registro por mail*, *Sistema de login con email y contraseña* y *Cerrar sesión*. Siendo importante pero no lo más prioritario de la primera release, se intentará implementar lo máximo posible de estas HU. No se acabarán totalmente en este *Sprint* porque representan muchos puntos de HU.

En las siguientes *releases* iremos ampliando de forma equilibrada las funcionalidades que nutren las búsquedas de juegos, las opciones relacionadas con las boxes y sus búsquedas y todo lo relacionado con los usuarios.

3. Reunión de planificación del sprint

El martes en la sesión del laboratorio comenzó la reunión de planificación del *Sprint* del grupo *TheVideoGameBox*. El primer paso fue presentar el *Sprint Goal* por parte del *Product Owner* (Julián). Las Historias de Usuario escogidas por Julián para estar en el *Sprint Goal* están registradas en la *tabla 3.1.1*.

Historias de Usuario
TVGB-S1: Búsqueda de juegos por nombre
TVGB-W1: Ver atributos básicos del juego
TVGB-L1: Sistema de registro por mail
TVGB-L2: Sistema de login con email/contraseña
TVGB-L3: Cerrar Sesión
TVGB-B1: Crear Box
TVGB-B2: Añadir juego a la box
TVGB-W2: Ver juegos de la box
TVGB-S2: Búsqueda de boxes por nombre

Tabla 3.1.1 Historias de Usuario del primer Sprint

Estas *Historias de Usuarios* (HU) están colocadas por orden de implementación, de forma que si a lo largo del *Sprint* hay alguna historia que no dé tiempo a hacer serán las últimas historias las que no se acaben.

Después de presentar el *Sprint Goal* el equipo de desarrollo expuso sus preguntas al *Product Owner* sobre las HU que causaban más dudas. Las preguntas que se expusieron fueron las que se encuentran en la *tabla 3.1.2*.

Preguntas expuestas en la reunión
¿Cuáles son los atributos básicos de un juego?
¿Qué datos se van a recoger en el registro por mail?
¿Cómo quiere el Product Owner el sistema de inicio de sesión?

Tabla 3.1.2 Preguntas Expuestas en la reunión

Después se empezó por definir que era un punto de historia en nuestro proyecto, para ello el *Equipo de Desarrollo* debatió entre ellos hasta llegar a la conclusión de que un punto de historia equivale a crear una vista de una funcionalidad.

3.1. Sistema de votación

Después de elegir a lo que equivale un punto de historia el *Scrum Master* indicó la escala que se iba a utilizar para estimar, sería la [escala de Cohn](#) por ser la más utilizada en el mundo real según los apuntes, acto seguido se comenzó a estimar los puntos de historia para cada *HU* con el método de [Planning Poker](#).

El *Scrum Master* escogió este método por ser el más interactivo ya que entre votación y votación habría debates entre el equipo de desarrollo para razonar la puntuación que habían dado a cada *HU*. Con este método todos los integrantes del equipo de desarrollo podrían dar su punto de vista sobre una *HU* y así el resto del equipo conocerá las dificultades que no han visto de primera mano. Las votaciones obtenidas por cada tarea se encuentran en la [tabla 3.1.3](#), en la cual se explica la votación final obtenida y cuantas veces hizo falta repetirla, y el *Sprint Goal* obtenido tras las votaciones.

Después de las votaciones en el *Equipo de Desarrollo* determinó junto al *Product Owner* que el *Sprint Goal* debería contener las *HU* con sus respectivas explicaciones de la [tabla 3.1.4](#).

Aunque al no tener experiencia previa este *Sprint Goal* seguramente no se cumpla y solo debería tomarse como una aproximación.

El diseño a alto nivel de las funcionalidades que se van a realizar durante el *Sprint* se encuentra en la página [Imagen 3.1.5](#).

Historias de Usuario	Votación obtenida
TVGB-S1: Búsqueda de juegos por nombre	En la primera votación hubo discrepancias, se tuvo que repetir una vez más para en una segunda votación decidir que esa HU tenía un valor de 2 puntos de historia.
TVGB-W1: Ver atributos básicos del juego	En la primera votación se decidió que esa HU tenía un valor de 2 puntos de historia.
TVGB-L1: Sistema de registro por mail	En la primera votación hubo discrepancias, se tuvo que repetir dos veces más para en una tercera votación decidir que esa HU tenía un valor de 8 puntos de historia.
TVGB-L2: Sistema de login con email/contraseña	En la primera votación se decidió que esa HU tenía un valor de 20 puntos de historia.
TVGB-L3: Cerrar Sesión	En la primera votación hubo discrepancias, se tuvo que repetir una vez más para en una segunda votación decidir que esa HU tenía un valor de 5 puntos de historia.
TVGB-B1: Crear Box	En la primera votación hubo discrepancias, se tuvo que repetir dos veces más para en una tercera votación

	decidir que esa HU tenía un valor de 3 puntos de historia.
TVGB-B2: Añadir juego a la box	En la primera votación hubo discrepancias, se tuvo que repetir una vez más para en una segunda votación decidir que esa HU tenía un valor de 8 puntos de historia.
TVGB-W2: Ver juegos de la box	En la primera votación se decidió que esa HU tenía un valor de 3 puntos de historia.
TVGB-S2: Búsqueda de boxes por nombre	En la primera votación se decidió que esa HU tenía un valor de 1 punto de historia.

Tabla 3.1.3 Votaciones

Después de las votaciones en el *Equipo de Desarrollo* determinó junto al *Product Owner* que el *Sprint Goal* debería contener las HU con sus respectivas explicaciones de la [tabla 3.1.4](#).

Aunque al no tener experiencia previa este *Sprint Goal* seguramente no se cumpla y solo debería tomarse como una aproximación.

El diseño a alto nivel de las funcionalidades que se van a realizar durante el *Sprint* se encuentra en la página [Imagen 3.1.5](#).

Historias de Usuario	Explicación
TVGB-S1: Búsqueda de juegos por nombre	Es la función principal de nuestra aplicación y por tanto es la primera que debe estar en el Sprint Goal.
TVGB-W1: Ver atributos básicos del juego	Creemos que lo más obvio una vez encontrado un juego en una búsqueda es poder ver sus atributos principales.
TVGB-L1: Sistema de registro por mail	El resto de las funciones que no son encontrar juegos solo las pueden realizar usuarios registrados, por tanto para seguir el orden de desarrollo y uso de la aplicación lo más lógico es continuar con el sistema de registro por mail.
TVGB-L2: Sistema de login con email/contraseña	Una vez registrado un usuario el siguiente paso es que se pueda loguear para entrar en su cuenta, por eso esta HU viene después de TVGB-L1: Sistema de registro por mail.
TVGB-L3: Cerrar sesión	Creemos que esta HU debe estar en el Sprint Goal para cerrar el ciclo de creación-registro y cierre de cuenta del usuario.
TVGB-B1: Crear Box	Una vez podemos buscar juegos es ideal que podamos crear una box a la cual añadir los juegos.

Tabla 3.1.4 Sprint Goal

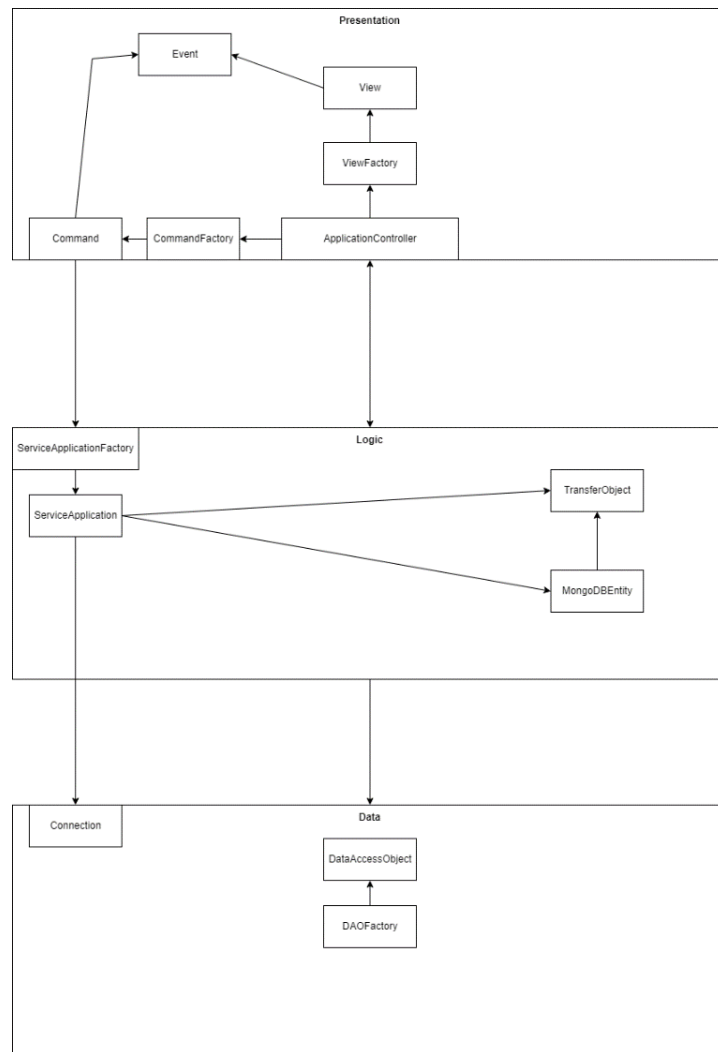


Imagen 3.1.5 Diseño de funcionalidades

3.2. Descomposición de las HU en tareas

El *Equipo de Desarrollo* descompuso las HU en tareas, por falta de tiempo en la reunión de planificación esta descomposición se realizó más tarde en una reunión online por parte del equipo de desarrollo. Al principio del *Sprint* todas las tareas aparecen en el estado *to do* en nuestra herramienta *Jira*. Empezaron haciendo un análisis técnico de cada HU y definiendo las clases que se iban a utilizar en cada tarea además de los test que iban a tener que pasar después. Para definir las clases y el análisis se utilizó la definición de hecho de nuestro proyecto. Como dentro del *Equipo de Desarrollo* hay personas con distinto nivel de experiencia creemos que vamos a necesitar utilizar *Swarming* cada vez que alguien se vea atascado en una HU. También creemos que el patrón *Rey-Sirviente* nos ayudará a no interrumpir a las personas que están haciendo una tarea de mayor prioridad que la de la persona que necesita ayuda. El objetivo principal de este *Sprint* es coger experiencia con scrum y aprender tanto a estimar de forma correcta como a trabajar en equipos con metodologías ágiles.

4. Sprint Backlog

Durante la segunda parte de la reunión de planificación el *Equipo de Desarrollo (ED)* se encargó de descomponer las *HU* del *Sprint Goal* de este primer *Sprint* en tareas más pequeñas, recopilándolas en el *Sprint Backlog*.

En un inicio, se realizó un esbozo rápido de la arquitectura principal de la aplicación y de las funcionalidades que el *ED* se comprometió a realizar en la primera parte de la reunión. Este diseño conceptual se puede ver en la *Imagen 3.1.5*.

Después de realizar el diseño conceptual, el *ED* empezó a descomponer las *HU* en distintas tareas, basándose en el diseño realizado, la arquitectura multicapa (*Presentación*, *Negocio* e *Integración*) y las entidades del modelo de dominio de la base de datos (*Game*, *Box* y *User*) que se puede ver en la *Imagen 4.1*. Las tareas iniciales en las que se descompusieron las *HU* se encuentran en la *Tabla 4.2*.

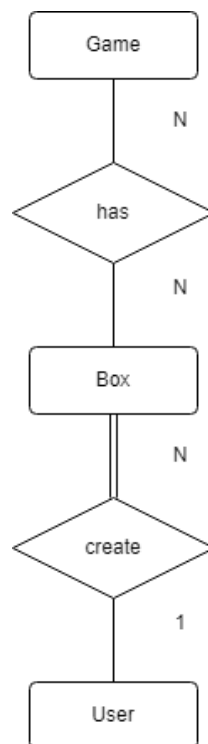


Imagen 4.1 Modelo de Dominio

Historias de Usuario	Tareas
Búsqueda de juegos por nombre	Hacer transfer de Game
	Añadir códigos de eventos a Event
	Crear entidad e MongoDB
	Hacer la función "SearchAllByName" en el DAO
	Hacer el comando particular "SearchAllByName" de Game

	Hacer la vista principal
	Hacer la vista particular de "SearchAllByName" de Game
Ver atributos básicos del juego	Hacer la función "SearchOne" en el DAO de Game
	Hacer la función "SearchOne" en el SA de Game
	Añadir código de eventos a Event
	Crear comando particular "SearchOne" de Game
	Hacer la vista particular "SearchOne" de Game
Crear Box	Hacer transfer de Box
	Añadir eventos a Event
	Hacer comando particular de "Create" de la box
	Hacer la función "Create" en el DAO de Box
	Hacer la función "Create" en el SA de Box
Añadir un juego a una Box	Hacer la vista particular de "Create" de Box
	Añadir evento a Event
	Hacer "AddGame" en el DAO de Box
	Hacer "AddGame" en el SA de Box
	Hacer el comando particular "AddGame" de Box
Búsqueda de boxes por nombre	Hacer la vista de la Box
	Hacer la vista particular "AddGame" de Game
	Añadir evento a Event
	Hacer la función "SearchAllByName" en el DAO de Box
	Hacer la función "SearchAllByName" en el SA de Box
Ver juegos de la Box	Hacer el comando particular "SearchAllByName" de Box
	Hacer la vista particular "SearchAllByName" la vista de Box
	Añadir evento a Event
	Hacer la función de "ShowGames" en el DAO de box
	Hacer la función "ShowGames" en el SA de Box
Sistema de registro por mail	Hacer el comando particular "ShowGames" en Box
	Hacer la vista particular "ShowGames" en la vista de Box
	Crear transfer de User
	Añadir la función "SignUp" en el DAO de User
	Añadir la función "SignUp" en el SA de User
Sistema de login con email/contraseña	Hacer el comando particular "SignUp" de Game
	Hacer la vista particular de "SignUp"
	Añadir la función "LogIn" en el DAO de User
	Añadir la función "LogIn" en el SA de User
	Hacer el comando particular "LogIn" de User
Cerrar Sesión	Hacer la vista particular de "LogIn"
	Hacer la vista principal de usuario loggeado
	Cambiar booleano "logged"

Tabla 4.2 Sprint Inicial

Tras añadir las tareas iniciales, se realizó un segundo diseño, más detallado que el diseño conceptual inicial, que se puede observar en la [Imagen 4.3](#), para tener la certeza de que no faltaba ninguna tarea relacionada con la arquitectura de la aplicación.

Fix versions:	None
----------------------	------

Type:	Historia	Priority:	High
Reporter:	José Díaz	Assignee:	Gonzalo Vilchez Rodriguez
Resolution:	Done	Votes:	0
Labels:	Dueño, NoRegistrado, Registrado		
Σ Remaining Estimate:	Not Specified	Remaining Estimate:	Not Specified
Σ Time Spent:	Not Specified	Time Spent:	Not Specified
Σ Original Estimate:	Not Specified	Original estimate:	Not Specified

Sub-tasks:	Key	Summary	Type	Status	Assignee
	TVGB-137	Añadir evento a Event	Subtarea	Done	David Elías Piñeiro
	TVGB-138	Hacer la funcion "SearchAllByName" en...	Subtarea	Done	JUAN ROMO IRIBARREN
	TVGB-139	Hacer la funcion "SearchAllByName" en...	Subtarea	Done	Alvaro Gomez Sittima
	TVGB-140	Hacer el comando particular "SearchAl...	Subtarea	Done	Carlos Carnero
	TVGB-141	Hacer la vista particular "SearchAllB...	Subtarea	Done	Carlos Carnero
	TVGB-161	Hacer pruebas de integracion	Subtarea	Done	Alvaro Gomez Sittima
Criterios de Aceptación:	<ol style="list-style-type: none"> 1. Cuándo un usuario quiera buscar boxes por nombre deberá introducir el nombre de la box en el buscador. 2. El usuario podrá introducir un máximo de 50 caracteres para realizar la búsqueda. 3. Cuando se realiza la búsqueda, si existen boxes que contengan parcial o totalmente el nombre introducido por el usuario, se listarán en orden inverso de creación (desde las más recientes a las más antiguas). 4. Cuando se realiza la búsqueda, si no hay ningún box con el nombre introducido, se mostrará un mensaje informando al usuario de lo sucedido. 				
Sprint:	Release 1				
Story point estimate:	1				
Rank:	0 00084:				

Description
Como jugador quiero poder buscar boxes mediante el nombre para encontrar rápidamente las que me interesen más.

[TVGB-82] TVGB-S1: Búsqueda de juegos por nombre Created: 27/Feb/22 Updated: 27/Mar/22 Resolved: 27/Mar/22

Status:	Done
Project:	TheVideoGameBox
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Historia	Priority:	Highest
Reporter:	José Díaz	Assignee:	Mario Campos Sobrino
Resolution:	Done	Votes:	0
Labels:	Dueño, NoRegistrado, Registrado		
Σ Remaining Estimate:	Not Specified	Remaining Estimate:	Not Specified
Σ Time Spent:	Not Specified	Time Spent:	Not Specified
Σ Original Estimate:	Not Specified	Original estimate:	Not Specified

Sub-tasks:	Key	Summary	Type	Status	Assignee
	TVGB-99	Añadir códigos de eventos a Event	Subtarea	Done	Javier de Vicente
	TVGB-100	Hacer la funcion "SearchAllByName" en...	Subtarea	Done	Javier de Vicente
	TVGB-101	Hacer la funcion "SearchAllByName" en...	Subtarea	Done	Javier de Vicente
	TVGB-102	Hacer la vista principal de Game	Subtarea	Done	Carlos Gómez López
	TVGB-103	Hacer la vista particular de "SearchA...	Subtarea	Done	Carlos Gómez López
	TVGB-105	Hacer el comando particular "SearchAl...	Subtarea	Done	Javier de Vicente
	TVGB-106	Crear entidad de MongoDB	Subtarea	Done	Alvaro Gomez Sittima
	TVGB-147	Hacer transfer de Game	Subtarea	Done	Alvaro Gomez Sittima
	TVGB-148	Hacer pruebas de integracion	Subtarea	Done	Alvaro Gomez Sittima
	TVGB-150	Hacer arreglos a la logica de negocio	Subtarea	Done	Alvaro Gomez Sittima
	TVGB-156	Arreglo de la vista SearchAllByName	Subtarea	Done	Carlos Gómez López
Criterios de Aceptación:	<ol style="list-style-type: none"> 1. Cuando un usuario quiera buscar juegos mediante el nombre deberá introducir en el buscador el nombre del juego que le interese. 2. El usuario podrá introducir un máximo de 50 caracteres para realizar la búsqueda. 				

	<ol style="list-style-type: none"> Cuando se realiza la búsqueda, si hay juegos que contienen parcial o totalmente en el nombre lo indicado por el usuario se mostrarán por pantalla. Cuando se realiza la búsqueda, si no hay ningún juego con el nombre introducido, se mostrará un mensaje informando al usuario de lo sucedido.
Sprint:	Release 1
Story point estimate:	2
Rank:	0 i00026:

Description

Como jugador quiero poder buscar juegos mediante el nombre para encontrar rápidamente aquellos en los que esté interesado y ver sus características de forma detallada.

[TVGB-80] TVGB-B2: Añadir un juego a un box Created: 27/Feb/22 Updated: 28/Mar/22 Resolved: 28/Mar/22

Status:	Done
Project:	TheVideoGameBox
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Historia	Priority:	High
Reporter:	José Díaz	Assignee:	Javier de Vicente
Resolution:	Done	Votes:	0
Labels:	Dueño		
Σ Remaining Estimate:	Not Specified	Remaining Estimate:	Not Specified
Σ Time Spent:	Not Specified	Time Spent:	Not Specified
Σ Original Estimate:	Not Specified	Original estimate:	Not Specified

Sub-tasks:	Key	Summary	Type	Status	Assignee
	TVGB-132	Añadir evento a Event	Subtarea	Done	Javier de Vicente
	TVGB-133	Hacer AddGame en el DAO de Box	Subtarea	Done	Javier de Vicente
	TVGB-134	Hacer AddGame en el SA de Box	Subtarea	Done	Javier de Vicente
	TVGB-135	Hacer el comando particular "AddGame"...	Subtarea	Done	Javier de Vicente
	TVGB-136	Hacer la vista particular "AddGame" d...	Subtarea	Done	JUAN ROMO IRIBARREN

	TVGB-153	Hacer pruebas de integracion	Subtarea	Done	Javier de Vicente
	TVGB-154	Hacer la vista de la box	Subtarea	Done	JUAN ROMO IRIBARREN
Criterios de Aceptación:	1. Cuando el usuario quiera añadir un juego a una box, si tiene boxes de su propiedad, podrá hacerlo desde las opciones de la box mediante añadir un juego, escogiendo de entre los juegos que tiene y/o los existentes en la aplicación.				
Sprint:	Release 1				
Story point estimate:	8				
Rank:	0 0007f:				

Description

Como dueño de una box quiero poder añadirle juegos para tenerlos organizados como desee.

[TVGB-75] TVGB-W2: Ver juegos de la box

Created: 27/Feb/22 Updated: 29/Mar/22 Resolved: 29/Mar/22

Status:	Done
Project:	TheVideoGameBox
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Historia	Priority:	High
Reporter:	José Díaz	Assignee:	Mario Campos Sobrino
Resolution:	Done	Votes:	0
Labels:	Dueño, NoRegistrado, Registrado		
Σ Remaining Estimate:	Not Specified	Remaining Estimate:	Not Specified
Σ Time Spent:	Not Specified	Time Spent:	Not Specified
Σ Original Estimate:	Not Specified	Original estimate:	Not Specified

Sub-tasks:	Key	Summary	Type	Status	Assignee
	TVGB-142	Añadir evento a Event	Subtarea	Done	Mario Campos Sobrino
	TVGB-143	Hacer la funcion de "ShowGames" en el...	Subtarea	Done	Javier de Vicente
	TVGB-144	Hacer la funcion "ShowGames" en el SA...	Subtarea	Done	Javier de Vicente

	TVGB-145	Hacer el comando particular "ShowGame..."	Subtarea	Done	Javier de Vicente
	TVGB-146	Hacer la vista particular "ShowGames"...	Subtarea	Done	Javier de Vicente
	TVGB-160	Hacer pruebas de integracion	Subtarea	Done	Javier de Vicente
Criterios de Aceptación:	<ol style="list-style-type: none"> 1. El usuario, mediante la opción ver juegos de la box, podrá ver los juegos que contiene la box con sus respectivos nombres y portadas. Los juegos se listan en orden inverso al que fueron añadidos, mostrando primero al usuario los juegos que son más recientes en la box. 2. Cuando se quieren ver los juegos de la box, si no hay ningún juego, aparecerá un mensaje informándolo. 				
Sprint:	Release 1				
Story point estimate:	3				
Rank:	0 i00085:				

Description

Como jugador quiero tener la posibilidad de ver el nombre y la portada de todos los juegos pertenecientes a una box que me interesa.

[TVGB-74] TVGB-B1: Crear box Created: 27/Feb/22 Updated: 27/Mar/22 Resolved: 27/Mar/22

Status:	Done
Project:	TheVideoGameBox
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Historia	Priority:	High
Reporter:	José Díaz	Assignee:	JUAN ROMO IRIBARREN
Resolution:	Done	Votes:	0
Labels:	Dueño, Registrado		
Σ Remaining Estimate:	Not Specified	Remaining Estimate:	Not Specified
Σ Time Spent:	Not Specified	Time Spent:	Not Specified
Σ Original Estimate:	Not Specified	Original estimate:	Not Specified

Issue links:	Relates relates to TVGB-79 TVGB-B7: Borrar box To Do				
Sub-tasks:	Key	Summary	Type	Status	Assignee
	TVGB-125	Hacer transfer de box	Subtarea	Done	David Elías Piñeiro

	TVGB-126	Añadir eventos a Event	Subtarea	Done	David Elías Piñeiro
	TVGB-127	Hacer la funcion "Create" en el DAO d...	Subtarea	Done	David Elías Piñeiro
	TVGB-128	Hacer la funcion "Create" en el SA de...	Subtarea	Done	David Elías Piñeiro
	TVGB-151	Hacer pruebas de integracion	Subtarea	Done	Alvaro Gomez Sittima
	TVGB-158	Hacer la vista particular de "Create"...	Subtarea	Done	David Elías Piñeiro
	TVGB-159	Hacer el comando particular de "Creat...	Subtarea	Done	David Elías Piñeiro
Criterios de Aceptación:	<ol style="list-style-type: none"> 1. Cuando un usuario quiera crear una box tendrá que obligatoriamente introducir el nombre, su descripción y la categoría/s a la que pertenece. 2. No se podrá crear la box si el título está vacío, tiene más de 50 caracteres o ya existe una box con ese título para ese usuario. Se informará de tal incorrección con un mensaje si se produce. 3. No se podrá crear la box si la descripción sobrepasa los 250 caracteres. Se informará de tal problema con un mensaje si se produce. 4. Cuando se está creando una box, si no se ha seleccionado al menos una categoría, no se podrá concretar la creación de la box. Se informará de tal problema con un mensaje. 5. Si la creación de la box se realiza correctamente, se informará al usuario mediante un mensaje. 				
Sprint:	Release 1				
Story point estimate:	3				
Rank:	0 i0006v:				

Description

Como jugador registrado quiero poder crear una box para organizar los juegos que tengo y/o me interesan en colecciones.

[TVGB-58] TVGB-W1: Ver atributos básicos del juego Created: 26/Feb/22 Updated: 27/Mar/22 Resolved: 27/Mar/22

Status:	Done		
Project:	TheVideoGameBox		
Components:	None		
Affects versions:	None		
Fix versions:	None		
Type:	Historia	Priority:	Highest
Reporter:	José Díaz	Assignee:	Alvaro Gomez Sittima
Resolution:	Done	Votes:	0
Labels:	Dueño, NoRegistrado, Registrado		

Σ Remaining Estimate:	Not Specified	Remaining Estimate:	Not Specified
Σ Time Spent:	Not Specified	Time Spent:	Not Specified
Σ Original Estimate:	Not Specified	Original estimate:	Not Specified

Sub-tasks:	Key	Summary	Type	Status	Assignee
	TVGB-107	Hacer la vista particular "ShowOne" d...	Subtarea	Done	JUAN ROMO IRIBARREN
	TVGB-108	Hacer la funcion "SearchOne" en el DA...	Subtarea	Done	JUAN ROMO IRIBARREN
	TVGB-109	Hacer la funcion "SearchOne" en el SA...	Subtarea	Done	Javier de Vicente
	TVGB-110	Añadir codigo de eventos a Events	Subtarea	Done	Javier de Vicente
	TVGB-111	Crear comando particular "SearchOne" ...	Subtarea	Done	Javier de Vicente
	TVGB-149	Hacer pruebas de integracion	Subtarea	Done	Javier de Vicente
	TVGB-155	Refactorizar game para añadir "fecha"	Subtarea	Done	Javier de Vicente
	TVGB-157	Arreglar vista ShowOne	Subtarea	Done	Carlos Gómez López
Criterios de Aceptación:	<ol style="list-style-type: none"> 1. Cuando un jugador quiera ver todos los atributos básicos de un juego, se le mostrará toda la información del juego: nombre, portada, compañías, géneros que definen a los juegos, plataformas y descripción. 2. Cuando un jugador elija la opción de ver detalles de un juego, si falta información de algún atributo, se mostrará un mensaje al usuario que indique que no hay información. 				
Sprint:	Release 1				
Story point estimate:	2				
Rank:	0 i0002e:				

Description

Como jugador quiero poder ver los atributos básicos de un juego para tener una idea general sobre sus características.

[TVGB-27] TVGB-L1: Sistema de registro por mail Created: 23/Feb/22 Updated: 29/Mar/22 Resolved: 29/Mar/22

Status:	Done
Project:	TheVideoGameBox
Components:	None
Affects versions:	None

Fix versions:	None
----------------------	------

Type:	Historia	Priority:	High
Reporter:	José Díaz	Assignee:	Carlos Gómez López
Resolution:	Done	Votes:	0
Labels:	NoRegistrado		
Σ Remaining Estimate:	Not Specified	Remaining Estimate:	Not Specified
Σ Time Spent:	Not Specified	Time Spent:	Not Specified
Σ Original Estimate:	Not Specified	Original estimate:	Not Specified

Issue links:	Blocks				
	blocks	TVGB-33	TVGB-L2: Sistema de login con email/c...		To Do
	blocks	TVGB-40	SRC: Sistema de recuperación de contr...		To Do
Sub-tasks:	Key	Summary	Type	Status	Assignee
	TVGB-112	Crear transfer de User	Subtarea	Done	Carlos Carnero
	TVGB-113	Añadir la funcion "SingUp" en el DAO ...	Subtarea	Done	Carlos Gómez López
	TVGB-114	Añadir la funcion "SingUp" en el SA d...	Subtarea	Done	JUAN ROMO IRIBARREN
	TVGB-116	Hacer el comando particular "SingUp" ...	Subtarea	Done	Carlos Gómez López
	TVGB-118	Hacer la vista particular de "SingUp"	Subtarea	Done	JUAN ROMO IRIBARREN
	TVGB-162	Hacer pruebas de integracion	Subtarea	Done	Gonzalo Vilchez Rodriguez
Criterios de Aceptación:	<ol style="list-style-type: none"> 1. Cuando un usuario quiera registrarse en la aplicación se le solicitará un email, un usuario y la contraseña que deberá confirmar para evitar posibles errores. 2. Cuando un usuario se está registrando en la aplicación, si introduce un mail existente saltará un mensaje de error comunicándole que ese email ya está en uso y deberá introducir uno nuevo. 3. Cuando un usuario se esta registrando en la aplicación, si introduce un usuario existente saltará un mensaje de error comunicándole que ese usuario ya existe y deberá introducir uno nuevo. 4. El nombre de usuario deberá tener como máximo 50 caracteres. 5. Cuando un usuario se esta registrando en la aplicación, si introduce una contraseña diferente para confirmar la contraseña puesta anteriormente, saltará un mensaje de error comunicándole el fallo y deberá volver a introducir las contraseñas de nuevo. 6. Cuando el usuario no registrado introduce bien todos los campos, se le creará un nuevo perfil para empezar a usar la aplicación como usuario registrado. 				
Sprint:	Release 1				

Story point estimate:	8
Rank:	0 i0008e:

Description

Como jugador no registrado quiero disponer de un sistema de registro en la aplicación mediante email y creación de contraseña, permitiéndome crear un perfil y poder disfrutar de las funcionalidades para usuarios registrados.

[TVGB-96] Sistema de Boxes <small>Created: 17/Mar/22 Updated: 29/Mar/22</small>	
Status:	To Do
Project:	TheVideoGameBox
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Epic	Priority:	High
Reporter:	José Díaz	Assignee:	José Díaz
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Sprint:	Release 1, Release 2
Issue color:	purple
Rank:	0 i000bf:

[TVGB-95] Sistema de Registro/Login/Cierre Sesión <small>Created: 17/Mar/22 Updated: 29/Mar/22 Due: 29/Mar/22</small>	
Status:	To Do
Project:	TheVideoGameBox
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Epic	Priority:	High
Reporter:	José Díaz	Assignee:	José Díaz
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		

Time Spent:	Not Specified
Original estimate:	Not Specified

Sprint:	Release 1, Release 2
Start date:	15/Mar/22
Issue color:	purple
Rank:	0 i000b7:

[TVGB-67] TVGB-L3: Cerrar sesión Created: 26/Feb/22 Updated: 29/Mar/22	
Status:	To Do
Project:	TheVideoGameBox
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Historia	Priority:	High
Reporter:	José Díaz	Assignee:	Alvaro Gomez Sittima
Resolution:	Unresolved	Votes:	0
Labels:	Dueño, Registrado		
Σ Remaining Estimate:	Not Specified	Remaining Estimate:	Not Specified
Σ Time Spent:	Not Specified	Time Spent:	Not Specified
Σ Original Estimate:	Not Specified	Original estimate:	Not Specified

Sub-tasks:	Key	Summary	Type	Status	Assignee
	TVGB-124	Cambiar booleano "logged"	Subtarea	To Do	José Díaz
Criterios de Aceptación:	<ol style="list-style-type: none"> 1. Cuando un usuario que previamente haya iniciado sesión (registrado), si ha terminado de utilizar la aplicación y desea salir de su cuenta, podrá hacerlo mediante una opción de cerrar sesión. 2. Si el usuario no ha iniciado sesión previamente no se mostrará en ningún momento la opción de cerrar sesión. 				
Sprint:	Release 1, Release 2				
Story point estimate:	5				
Rank:	0 i00098:				
Flagged:	Impediment				

Description
Como jugador registrado, quiero poder cerrar sesión para finalizar el uso de la aplicación mediante mi usuario.

[TVGB-33] TVGB-L2: Sistema de login con email/contraseña

Created: 23/Feb/22 Updated: 29/Mar/22

Status:	To Do
Project:	TheVideoGameBox
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Historia	Priority:	High
Reporter:	José Díaz	Assignee:	JUAN ROMO IRIBARREN
Resolution:	Unresolved	Votes:	0
Labels:	Dueño, Registrado		
Σ Remaining Estimate:	Not Specified	Remaining Estimate:	Not Specified
Σ Time Spent:	Not Specified	Time Spent:	Not Specified
Σ Original Estimate:	Not Specified	Original estimate:	Not Specified

Issue links:	Blocks
	is blocked by TVGB-27 TVGB-L1: Sistema de registro por mail Done

Sub-tasks:	Key	Summary	Type	Status	Assignee
	TVGB-119	Añadir la funcion "LogIn" en el DAO d...	Subtarea	To Do	José Díaz
	TVGB-120	Añadir la función "LogIn" en el SA de...	Subtarea	To Do	José Díaz
	TVGB-121	Hacer el comando particular "LogIn" d...	Subtarea	To Do	José Díaz
	TVGB-122	Hacer la vista particular de "LogIn"	Subtarea	To Do	José Díaz
	TVGB-123	Hacer la vista principal de usuario l...	Subtarea	To Do	José Díaz

Criterios de Aceptación:	<ol style="list-style-type: none"> 1. Cuando un usuario previamente registrado quiera iniciar sesión con su usuario y contraseña deberá seleccionar la opción login e introducir su usuario y contraseña. 2. Cuando el usuario introduce un email y una contraseña, si no se corresponden con un usuario registrado saltará un mensaje de error y el usuario deberá introducir sus datos de nuevo.
---------------------------------	--

Sprint:	Release 1, Release 2
Story point estimate:	20
Rank:	0 i0008o:
Flagged:	Impediment

Description

Como jugador registrado con un usuario y una contraseña quiero poder iniciar sesión en la aplicación con mis credenciales cuando quiera, para disfrutar de las funcionalidades de usuario registrado de mi perfil.

5. Desarrollo del Sprint

Para hablar del desarrollo del *Sprint* vamos a ir describiendo cómo ha ido trabajando el equipo a lo largo de la duración de este, en las dos semanas comprendidas entre el martes 15 de marzo y el martes 29 de marzo. Además, incluimos detalles como cuántas de las tareas del *Sprint Backlog* se consiguieron terminar, qué problemas aparecieron y cómo se solventaron.

Para comenzar el *Sprint* el *Equipo de Desarrollo* se reunió para decidir cómo trabajaríamos a la hora de codificar. Para implementar las distintas historias de usuario habíamos elegido ya previamente utilizar tanto *Eclipse* como *IntelliJ*. Dentro de estas dos, creamos una rama para cada historia de usuario en la que comenzábamos a trabajar, borrándola una vez que dicha historia de usuario se consideraba finalizada, y haciendo *pull-request* en cada una de ellas antes de empezar a trabajar para que se actualice el trabajo de los compañeros y no se pierda nada por el camino. Además, creamos una rama *dev* (la rama final), que engloba toda la parte del código que ya está finalizado y funciona.

Comenzamos cogiendo cada uno una tarea (siempre respetando la prioridad), la codificábamos, y al terminar con todas las tareas de una *Historia de Usuario* cerrábamos esa rama. Las últimas tareas siempre solían ser la vista y las pruebas, concretamente las últimas siempre son las pruebas ya que dependen de que todo lo demás esté listo. Siempre que alguien se asignaba una tarea, no pasaba a otra distinta hasta haber finalizado la suya. Al finalizar una *Historia de Usuario*, esta pasa siempre a estar pendiente de validación por parte del *Product Owner*, quién realiza las pruebas necesarias para validarla y poder incluir esa parte del código en la rama *dev*.

Nuestro primer problema vino con las ramas. Hubo una mala gestión de éstas, ya que se hicieron cambios que afectaban a todas las ramas de funcionalidad dentro de una rama de funcionalidad en concreto. Esto supuso que al hacer *merge* no se sabía con certeza cual era la versión más actualizada, se sobrescribían cambios o se mezclaban de forma incorrecta produciendo clases desactualizadas completamente o actualizadas sólo parcialmente. Todo esto supuso que el equipo de desarrollo tuviese que hacer una pausa y centrarnos todos en resolver esos conflictos y arreglar las ramas para poder seguir asignándonos nuevas tareas.

Otro de los problemas que tuvimos fue con el ejecutable. Tuvimos problemas a la hora de crear el primer ejecutable, ya que no permitía ejecutarlo si no se tenía el *JDK 17* instalado, por lo que tuvimos que bajar la versión del proyecto a *JDK 1.8* ya que esto permite ejecutar el programa solo con el *JRE* (normalmente

instalado en todos los ordenadores). Otra solución hubiese sido crear un instalador o incluir los archivos necesarios del *JDK* dentro del ejecutable, pero ninguna de estas soluciones era fácil de implementar ni considerábamos que fuese lo más adecuado para este proyecto.

En cuanto a las tareas del *Sprint Backlog*, conseguimos terminar todas las tareas relativas a las historias de usuario *búsqueda de juegos por nombre*, *ver atributos básicos del juego*, *crear box*, *añadir un juego a una box*, *búsqueda de boxes por nombre* y *ver juegos de la box*, todas ellas más detalladas en el *punto 4* (*Sprint Backlog*). Sin embargo, no terminamos todas las tareas ni de *sistema de login con email/contraseña* ni de *cerrar sesión*, como estimamos.

Con respecto a las pruebas realizadas para testear y validar el software desarrollado, hemos seguido parcialmente el plan de pruebas definido previamente en la anterior práctica. Empezando por las *pruebas unitarias*, como ya establecimos, las han realizado los encargados de codificar la funcionalidad a testear. Son pruebas automatizadas y realizadas con la herramienta *JUnit en su versión 4* en las que hemos comprobado si cada unidad software se comportaba de forma correcta y, si no era así, corregir los errores cometidos.

Tras haber realizado las pruebas unitarias de los diferentes módulos, el siguiente paso fue llevar a cabo las *pruebas de integración*, que son pruebas automatizadas cuyo fin es validar el funcionamiento de los distintos módulos de la aplicación como grupo e identificar los problemas que surgen cuando los distintos componentes se combinan entre sí. Los encargados de realizar las pruebas de integración, como en las pruebas unitarias, son aquellos integrantes del equipo de desarrollo que han contribuido en la implementación de los módulos a testear y se ha seguido la estrategia *Bottom-Up* (empezando por los componentes que no llaman a otras partes del programa y tras ellos, los componentes que llaman solo a la parte ya integrada).

Una vez completadas las pruebas de cada historia de usuario, el siguiente paso fue la realización de las *pruebas de aceptación* para validar la aplicación desde el punto de vista del usuario final. El responsable de realizarlas ha sido nuestro *Product Owner*, Julián Moffatt, ya que conoce bien la funcionalidad de la aplicación. El informe sobre las pruebas de aceptación elaborado por Julián es el desarrollado a continuación.

5.1 Primera Revisión

Realizando las pruebas de aceptación de *Buscar juegos por nombre*, todo se ejecutó como era esperado cuando se introdujo una búsqueda que coincide con alguno de los juegos existentes en la base de datos. Sin embargo, cuando la búsqueda no tiene resultados, no se informa al usuario como está especificado en los criterios de aceptación, siendo necesario corregirlo. Si en la búsqueda se introducen más de 50 caracteres, el comportamiento no es el esperado y desaparece la ventana. Se debe indicar al usuario de que no debe introducir tantos caracteres al realizar la búsqueda mediante un mensaje de error. Por otro lado, la vista tiene una apariencia visual buena y no es necesario mejorarla más. Deben arreglarse los errores de

funcionamiento que no concuerdan con los criterios de aceptación y la historia de usuario podrá ser validada.

En segundo lugar, realizando las pruebas de aceptación de *Ver atributos básicos del juego*, se debe sustituir el atributo *categoría* por *género*, siguiendo la actualización del término en el proyecto. La presentación de la vista debe mejorarse porque están presentados los atributos de forma desordenada y desalineada. Cambiando esto según los criterios y mejorando la apariencia de la vista la *HU* podrá ser validada.

Haciendo las pruebas de aceptación de *Crear box*, cuando se quiere crear la box no habiéndose completado todo campo obligatorio, se informa correctamente al usuario de que no puede crear la box. Sin embargo, no se indica al usuario que la box se ha creado correctamente cuando el formulario para crear la box está bien rellenado. No se permite al usuario elegir múltiples géneros como sí debería ser posible. La vista es estéticamente correcta. Será validada cuando se cumplan todos los criterios de aceptación.

Cuando el *Equipo de Desarrollo* informó al *Product Owner* de que los detalles que no se ajustaban a los criterios de aceptación habían sido subsanados, este repitió las pruebas de aceptación para validar las historias de usuario pendientes.

5.2 Segunda revisión

En *Buscar juegos por nombre*, ahora si se informa al usuario cuando no hay resultados para una búsqueda. También se muestra un mensaje de error cuando se introducen más de los 50 caracteres permitidos. Si es posible en el futuro, se debería intentar optimizar un poco más el tiempo de ejecución de la búsqueda. Estableciendo un número de caracteres mínimo, obligamos al usuario a hacer una búsqueda significativa que tendrá unos resultados suficientemente filtrados como para no lastrar los tiempos de ejecución. Esta vez la *HU* es validada porque es perfectamente funcional y estéticamente correcta.

Revisando la *HU Ver atributos básicos del juego*, se observó que se había añadido el atributo plataforma a la vista. También se actualizó el atributo *categoría* por *género* como correspondía. Se mejoró la presentación de los atributos, siendo más clara y ordenada. La *HU* es perfectamente funcional y estéticamente correcta, es validada.

En cuanto a la *HU Crear box*, ahora se indica al usuario cuando la box se crea correctamente mediante un mensaje de información. También se ha incluido la selección múltiple de géneros. La *HU* es validada porque cumple con los criterios de aceptación y estéticamente es correcta.

Una vez validadas las tres primeras *HU*, cuando el *Equipo de Desarrollo* tenía listas las siguientes *HU* se realizaron las pruebas de aceptación.

Se empezó con la realización de pruebas para *Buscar boxes por nombre*. Se observó que cuando se buscan boxes existentes en la base de datos funciona correctamente. Si no se obtienen resultados de la búsqueda, se informa al

usuario correctamente. Cuando se introducen más de 50 caracteres se informa al usuario de que no debe introducir tantos mediante un mensaje de error. La *HU* cumple los criterios de aceptación y estéticamente es correcta, es validada.

Respecto a la *HU* [Añadir juego a la box](#), se cumplen los criterios de búsqueda de la *HU* ya validada [búsqueda de juegos por nombre](#) ya que está reciclada en esta implementación. Realizando las pruebas se comprueba que se añaden correctamente los juegos a la box. La *HU* cumple con los criterios de aceptación y es estéticamente correcta. Es validada por el *Product Owner*.

Realizando las pruebas de aceptación de [Ver juegos de la box](#), se comprueba que se listan los juegos de la box correctamente y la presentación es estéticamente agradable. La *HU* es perfectamente funcional y estéticamente perfecta, es validada.

En cuanto a las pruebas de aceptación de [Sistema de Registro por email](#), se comprueba correctamente que las contraseñas que se introducen son idénticas y que se rellenan todos los campos de forma obligatoria. Se informa al usuario cuando se crea la cuenta adecuadamente. Sin embargo, no se comprueba que el correo introducido contenga un @ y por lo tanto no filtraría un correo totalmente inventado. La presentación de la vista es aceptable.

Finalmente, cuando el *Equipo de Desarrollo* corrigió el detalle de [Sistema de Registro por email](#) qué permitía correos que no contienen un @, el *Product Owner* repitió las pruebas y validó la *HU*.

Para permitir que todos los integrantes del *Equipo de Desarrollo* combinen correctamente cambios en el código del repositorio es necesario emplear [Integración Continua](#), un proceso automatizado que consiste en compilar y realizar las pruebas sobre el proyecto (tras detectar una acción determinada) que sirve para verificar si el nuevo código es correcto o en caso de que se produzca algún error, este se pueda corregir de forma más sencilla. Las acciones que activan el proceso de *Integración Continua* en nuestro proyecto son la realización de un [push](#) o un [pull-request](#) sobre la rama [stage](#). Tras detectar una de estas acciones, se compila el proyecto, se ejecutan las pruebas unitarias y de integración y si todo ha ido bien, se sube el proyecto compilado al repositorio como un artefacto y un ejecutable. Para ello hemos desarrollado un código que se encuentra en las imágenes [Imagen 5.3](#) e [Imagen 5.4](#).

```

1  name: TVGB_CI
2
3  on:
4    push:
5      branches:
6        - stage
7    pull_request:
8      branches:
9        - stage
10   workflow_dispatch:
11
12  jobs:
13    CI:
14      runs-on: ubuntu-latest
15
16      steps:
17        - name: Paso 1 - Checkout rama stage
18          uses: actions/checkout@v3
19          with:
20            ref: 'stage'
21
22        - name: Paso 2.1 - Preparar JDK 17
23          uses: actions/setup-java@v2
24          with:
25            distribution: 'temurin'
26            java-version: '17'
27            cache: 'maven'
28
29        - name: Paso 2.2 - Comprobar version de Java
30          run: java -version
31
32        - name: Paso 3.1 - Compilar/Build el proyecto de Java con Maven
33          run: |
34            cd Codigo/TheVideoGameBox
35            mvn clean -B package -Dmaven.test.skip=true
36            mkdir build && cp target/*.jar build
37
38        - name: Paso 3.2 - Comprobar que se ha compilado el proyecto correctamente
39          run: |
40            cd Codigo/TheVideoGameBox/build
41            ls | grep .jar
42

```

Imagen 5.3 Primera parte del código

```

43    - name: Paso 4.1 - Ejecutar tests unitarios
44      run: |
45        cd Codigo/TheVideoGameBox
46        mvn -Dtest="Unit/**" -DfailIfNoTests=false test
47
48    - name: Paso 4.2 - Ejecutar tests de integracion
49      if: success()
50      run: |
51        cd Codigo/TheVideoGameBox
52        mvn -Dtest="Integration/**" -DfailIfNoTests=false test
53
54    - name: Paso 5.1 - Subir el proyecto compilado como un artefacto
55      if: success()
56      uses: actions/upload-artifact@v2
57      with:
58        name: TheVideoGameBox_Executable
59        path: Codigo/TheVideoGameBox/build

```

Imagen 5.4 Segunda parte del código

Presentamos también dos ejemplos con resultados obtenidos tras realizarse la *integración continua* que se encuentran en las imágenes [Imagen 5.5](#) e [Imagen 5.6](#)

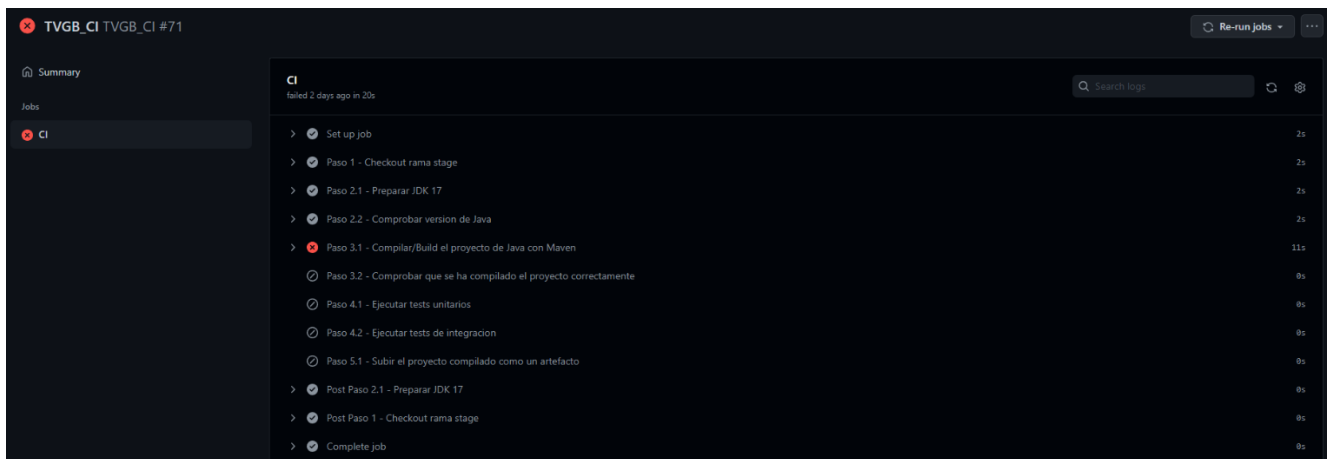


Imagen 5.5 Integración continua fallida

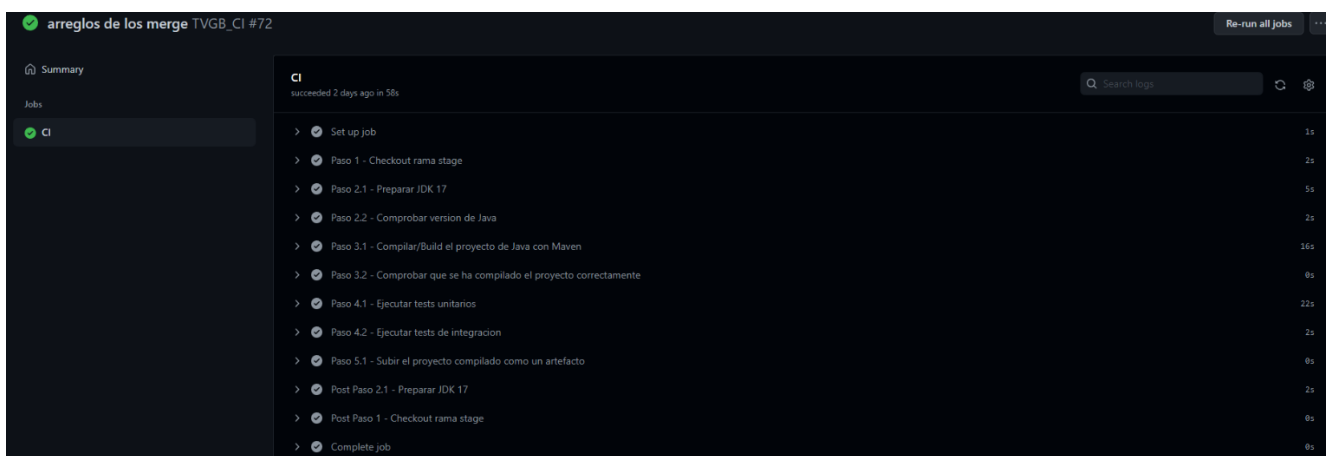


Imagen 5.6 Integración continua exitosa

6. Reuniones diarias

Se realizaron distintas reuniones diarias la cuales han servido al equipo para poder obtener información de lo que ha realizado cada integrante del Equipo de Desarrollo, como realizar preguntas al Product Owner o aclarar entre ellos algunos problemas o dudas generadas.

En total se llevaron 6 reuniones diarias, la cuales se situaron en unos días y hora concreta aprobada por todo el equipo. Estos días fueron: martes a las 12:40, jueves a las 19:00 y los sábados a la 13:00. Al situar estas fechas todos dimos nuestro aprobado, pero más adelante nos percatamos que el Sprint terminaba el lunes 28, por lo que la daily asignada al martes no se podría llegar a realizar por lo que decidimos cambiar dicho martes por lunes 28 a las 18:30.

A continuación se redactará un resumen de cada *daily scrum* realizada.

6.1. Daily scrum 17-03-2022

Puesto que es la primera reunión diaria nadie puede hablar de los problemas que ha encontrado dado que no ha comenzado el desarrollo, esta reunión nos sirve para coger experiencia y romper el hielo. Los integrantes del *Equipo de Desarrollo (ED)* solamente comentan las tareas que se han asignado y si han llegado a finalizar alguna de ellas.

Link reunión: <https://youtu.be/wejGN1-pPhY>

6.2. Daily scrum 20-03-2022

En esta reunión ya se comienza a ver al *ED* que ha empezado con sus tareas y algunos han tenido dudas o problemas técnicos generales, además uno de los compañeros sufrió problemas técnicos de conexión la hora de la reunión, pero pudo solucionarlos y continuar en ella.

Link reunión: <https://youtu.be/b8uK3cZYmFY>

6.3. Daily scrum 22-03-2022

La tercera reunión se hizo de forma presencial aprovechando que habíamos salido de clase todos y teníamos tiempo libre, el percance más notorio de la reunión es la lesión de un miembro del *ED*, a causa de eso hubo un retraso con las tareas pendientes. También se descubrió que hubo un parón en el producto, ya que surgieron problemas con el repositorio de *GitHub*, que requerían ser corregidos antes de continuar.

Link reunión: <https://youtu.be/1RyKLWMYGZQ>

6.4. Daily scrum 24-03-2022

En la reunión se habló de cómo hacer un ejecutable que haga posible no necesitar el *JDK*. También se comentó que se están retocando las *HU* para que pasen las pruebas de aceptación según las indicaciones del *Product Owner*. Finalmente se habló que se produjeron conflictos en *GitHub* y se habló sobre que continuaban sin resolverse, pero que ese mismo día o el viernes 25, ya estaría de nuevo operativo. Por último se hicieron diagramas de clase y vistas para las *HU*.

Link reunión: https://youtu.be/CZofoloZZ_M

6.5. Daily scrum 27-03-2022

Se consiguió hacer el ejecutable sin necesidad del *JDK* como se comentó en la anterior reunión daily. También no se pudo avanzar mucho porque se estaba resolviendo el problema de *GitHub*. Se están asignando varios DAO, SA y vistas para continuar avanzando. Por último se comentó que un *commit* de un integrante del equipo de desarrollo había generado errores para mostrar imágenes en búsquedas.

Link reunión: <https://youtu.be/eSfy0achm4U>

6.6. Daily scrum 28-03-2022

Se están retocando vistas, se está haciendo *DAO*, *SA*, eventos, preparando pruebas, etc. También se comentó que es un problema que haya cuatro historias de usuario abiertas y sin finalizar quedando poco tiempo para que acabe el sprint. Por último se habló de cómo mejorar la dinámica de trabajo haciendo vistas porque son tediosas y retrasan al equipo de desarrollo, proponiéndose la posibilidad de reciclar código o utilizar software especializado para hacer vistas fácilmente.

Link reunión: <https://youtu.be/kpRFJOZWqPM>

7. Backlog con Impedimentos

El *Impediment Backlog* se compone de todos aquellos elementos o situaciones que afectan de forma negativa al desarrollo del trabajo. A continuación se mostrarán las situaciones que hemos reunido que afectan a nuestro equipo.

7.1. Un miembro del equipo se va de viaje

Al comienzo del *Sprint* un miembro del *Equipo de Desarrollo* se va de viaje y no puede cumplir con sus tareas, se decide que como lo único que se puede modificar en el *Sprint* es el alcance se reducirá el *Sprint Goal*. Cuando vuelva del viaje continuará con las tareas pendientes.

7.2. Lesión de un miembro del equipo

Realizando una actividad deportiva uno de los miembros del *Equipo de Desarrollo* se lesiona y pierde un día de trabajo. Puesto que se había quedado a medias en una tarea de la cual dependen otras tareas se decide hacer *Swarming* con su tarea para acabarla mientras él no está disponible.

7.3. Dificultad para resolver algunas dudas con el Product Owner

Debido a la poca experiencia en *Scrum* de todos los miembros del equipo, el *Product Owner* solicita ayuda al *Equipo de Desarrollo* para ejecutar y probar la aplicación y mientras le ayudan dejan de progresar con las tareas.

7.4. Falta de formación

El hecho de implementar la base de datos en *MongoDB* hace que el *Equipo de Desarrollo* tenga que estudiar esa tecnología para poder empezar a hacer las tareas, esto alarga el estado *Doing* de cada tarea relacionada con la *Base de Datos*.

7.5. La red de la facultad bloquea el acceso a MONGODB

La red de la facultad bloquea el acceso a *MongoDB* y por tanto nuestra aplicación no tiene acceso a la base de datos, se decide utilizar los datos

móviles para poder tener acceso, los datos móviles van muy lentos debido al edificio de la facultad.

7.6. Distintos niveles de conocimiento técnico por parte del Equipo de Desarrollo

Debido a que hay distintos niveles de conocimiento sobre *Java* dentro del *Equipo de Desarrollo* no todos los miembros pueden trabajar a la misma velocidad lo que provoca retrasos, se intenta hacer *Swarming* para ayudar a los miembros del *Equipo de Desarrollo* más atascados.

8. Burndown Chart

El *Burndown Chart* es el indicador principal que informa sobre las posibilidades de alcanzar el compromiso adquirido por el equipo al finalizar el *Sprint*. Se trata de un gráfico indicativo de la cantidad de puntos de historia que se han acabado y de los que quedan por hacer contrastado con la cantidad de tiempo que ha pasado en el *Sprint*.

Como se puede apreciar en nuestro diagrama, *Imagen 8.1*, no forma una línea recta que empieza abajo del todo el primer día y acaba arriba del todo en el último día del *Sprint*, de hecho tiende a ser una recta que indica que los primeros días del *Sprint* no se acabó ninguna *HU*. Esto se debe a que es el primer *Sprint* y días la mayoría del *Equipo de Desarrollo* tuvo un examen de otra asignatura y por tanto no tuvieron tiempo de empezar con el proyecto.

A mitad del *Sprint* hubo un problema con las ramas del *SCV* que también retrasó al *Equipo de Desarrollo*. Los últimos días del *Sprint* fuimos acabando las *HU* y debido a un fallo en la comunicación con el *Product Owner* no le dijimos que podía empezar a validar las *HU* acabadas y por eso cuando se lo dijimos es cuando las empezó a validar, se puede ver cómo en ese momento el gráfico empieza a subir. Como habíamos previsto al comienzo del *Sprint* no hemos estimado bien los puntos de historia de cada *HU* ni los puntos de historia que somos capaces de realizar a lo largo del *Sprint*. Se puede observar que nos hemos quedado a la mitad de los puntos de historia que habíamos previsto terminar, está claro que con mayor experiencia y conocimiento sobre nuestro propio equipo vamos a ser capaces de estimar de forma más precisa.

Date - March 15, 2022 - March 29, 2022

Sprint goal - Máxima prioridad a las HU que implementan las funciones que se ofrecen a usuarios no registrados - Búsqueda de juegos por nombre - Ver atributos básicos del juego Se implementarán las primeras funciones que irán orientadas a usuarios registrados en versiones futuras, mediante las HU: - Crear box - Añadir un juego a la box - Búsqueda de boxes por nombre - Ver juegos de la box No estando implementado el bloque que engloba todo lo relacionado con logueo, se implementan de forma provisional sin estar ligadas al usuario. Se buscará implementar todo el sistema de logueo que forman las HU: - Sistema de registro por email - Sistema de login con email y contraseña - Cerrar sesión Probablemente por puntos de HU se acabará esto en el siguiente sprint.

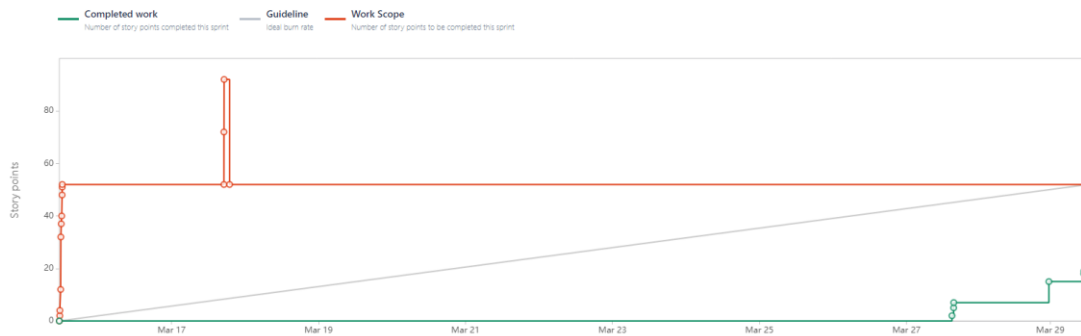


Imagen 8.1 Diagrama de Burndown Chart

9. Reunión de revisión

Comienza la reunión en el horario de clase con todo el equipo presente, para comenzar se revisa el producto por encima, conversando entre los miembros del equipo, sobre las vistas se llega a la conclusión de que tienen un aspecto algo anticuado, también se habla de que a la hora de registrar un usuario la contraseña no se encripta, y aunque no está en los criterios de aceptación se le recomienda incluirlo.

En la vista principal se le dice al *Product Owner (PO)* que resulta incómoda realizar las búsquedas y se proporciona una distribución de los elementos de la vista distinta a la que hay, por ejemplo, que la vista principal sea buscar juegos y al loguearte no aparezca la opción de buscar boxes sino que aparezcan las tuyas directamente. Los colores básicos de la aplicación parecen poco atractivos para la mayoría del *Equipo de Desarrollo (ED)* pero al *PO* le gustan y por tanto no se cambiarán.

A continuación se da un repaso de las funcionalidades del producto una por una y se valora la opinión del equipo quedando registradas en la [tabla 9.1](#).

La reunión finaliza con una duración de **50 minutos** y la participación de todo el equipo *Scrum*.

Vistas del producto	Opiniones otorgadas
Vista principal	Distribuir los elementos de búsqueda
	Se revisa la posibilidad de cambiar las 4 búsquedas que hay por un combo box donde se seleccione el tipo de búsqueda y un solo cuadro de texto donde meter la búsqueda
	Se ofrece la opción de cambiar de lado los botones de login, actualmente están a la izquierda y se propone situarlos a la derecha como en la mayoría de las aplicaciones

	Se propone quitar el botón de create box de esa esquina superior derecha
	El <i>PO</i> defiende su postura de mantener la vista como esta y añadir las funciones básicas de las boxes pero eso sobrecarga mucho la vista y la vuelve menos práctica
Vista principal sin estar logueado	Se propone que en lugar de como está ahora aparezcan los 5 juegos más populares además de un botón de ver más.
	Se le comenta al <i>PO</i> que se pueden poner juegos aleatorios, al <i>PO</i> le gusta esta idea
Create Box	No gusta la forma de elegir géneros que hay hecha y se buscan maneras de solucionarlo como poner un <i>combobox multiselección</i> en el caso de saber cómo hacerlo
	El resto de la <i>HU</i> parece correcto al <i>ED</i>
Búsqueda de juegos por nombre	En general está bien, se estudia la forma de hacer una búsqueda más eficiente para que no tarde mucho la consulta en la base de datos dado que parece un poco lenta
	Se estudia poner un mínimo de caracteres a la hora de buscar, un mínimo de 4 caracteres sería aceptable ya que no hay juegos con menos de 4 caracteres
Búsqueda de boxes por nombre	En general está bien
	Se estudia la forma de hacer una búsqueda más eficiente
	Poner un mínimo de caracteres a la hora de buscar, un mínimo de 4 caracteres que es el número mínimo de caracteres cuando se crea una box
Ver atributos del juego	Todo el equipo concuerda en que está a gusto con esta <i>HU</i> , tanto con la forma de funcionar de su lógica como con su aspecto visual
Añadir box	No se debería haber hecho dependiente de ver atributos de una box puesto que la segunda <i>HU</i> no pertenece a este <i>Sprint</i> , por tanto no se puede validar la información
	No tiene sentido dejar una <i>HU</i> a medias a posta en un <i>Sprint</i>

Tabla 9.1 Conclusiones obtenidas de la reunión

10. Reunión de retrospectiva

En esta reunión todo el *Equipo de Desarrollo (ED)* valoró el proceso que se ha seguido durante el *Sprint* para ir cumpliendo las tareas, además se estudiaron formas de mejorar este desarrollo.

10.1. Recopilación de información del Sprint

Para comenzar la reunión hemos hecho dos listas valorando los aspectos bien hechos y los aspectos mal hechos a lo largo del *Sprint*, hemos dividido en dos listas todas aquellas cosas que nos parecían relevantes.

10.1.1. Cosas bien realizadas

Hemos ido mejorando las *dailies* con una clara evolución cada vez eran más efectivas y sobre todo dinámicas, unos resolvían las dudas de los otros, la organización ha sido buena dentro de la reunión y resultaban productivas para el *ED* que lograba solucionar dudas y pedir ayuda al resto del equipo cuando era necesario.

Nunca ha faltado nadie a una reunión *daily*. Las relaciones personales son buenas, se logra una buena comunicación a través de *WhatsApp* y siempre se ayuda a las personas con las tareas más atrasadas.

10.1.2. Cosas mal realizadas

El horario de las *dailies* no es suficientemente consistente, alguna reunión sobre todo al principio tenía menos contenido.

Surgieron problemas de conocimiento sobre la herramienta *GitHub*, y se reconoce una mejora del uso de *GitHub* a lo largo del tiempo pero sigue siendo insuficiente para sacar partido a la herramienta.

Se produjo un parón de desarrollo por culpa de los errores con las ramas del *SCV* no debe repetirse.

Otro inconveniente fue debido a la diferencia de conocimientos sobre programación debemos intentar dejar las tareas más sencillas a las personas que tienen más dificultades aunque esto pueda retrasar el desarrollo.

También ha habido tareas que se quedan atrasadas más de 2 días y al tener otras tareas dependientes retrasan todo.

A veces se han empezado demasiadas *HU* sin haber acabado las anteriores lo que provocaba que otros miembros del *ED* no pudieran coger esas tareas al estar ya asignadas y que además no se avanzara con las tareas atrasadas Y derivando de la anterior se han cogido más de una tarea a la vez.

10.2. Objetivos para el próximo Sprint

Para el próximo *Sprint* se ha decidido mejorar distintos aspectos, empezando por mejorar la gestión de *SCV*, mejorar la organización individual y colectiva, e intentar que las *dailies* sean algo más productivas y concordar un horario para poder seguirlo.

También se ha propuesto mejorar la gestión de tareas en *Jira* y dar más prioridad a acabar una *HU* antes de empezar una nueva. Además a la hora de hacer

nuevas tareas antes revisar si hay alguna que lleve en el estado de *doing* mucho tiempo para contactar con su asignado y ofrecer ayuda.

10.3. Plan de mejora

Una vez que tenemos los objetivos del *Sprint* claros creamos un plan de mejora para llegar a cumplir esos objetivos.

Nuestro plan de mejora consiste en intentar mantener el nivel y la fluidez de las últimas reuniones diarias y mejorar la comunicación expresando sobre todo las dificultades a las que nos hemos enfrentado por si alguien tiene la solución a esa dificultad. También debemos dar más importancia en las reuniones *daily*, tanto contar lo que se va a hacer y dar un poco más de detalle sobre cómo se va a hacer.

En *GitHub* debemos fomentar la nueva creación de ramas por muy pequeña que sea la mejora que se quiera hacer, para cada mejora debe haber obligatoriamente una rama nueva. Siguiendo con el *SCV* todas las ramas deben partir de *DEV* y antes de implementar una *HU* se debe unificar todo en *DEV* por tanto al acabar una *HU* se debe comunicar al resto del equipo para unificar las ramas y que no surjan conflictos. Tenemos que buscar la forma de que cosas comunes se puedan subir por separado al *DEV* para no generar conflictos entre otras funcionalidades.

Se impusieron también una serie de reglas de equipo a cumplir para que no vuelvan a surgir problemas como en el primer *Sprint*. Estas normas son las dos siguientes:

- Hacer siempre *pull-request* a la rama principal.
- Hay ciertas tareas como eventos y *transfers* que se deben hacer el mismo día que se asignan ya que son sencillas y rápidas de completar, y otras como las vistas que se puede tardar más.

Por último, vamos a hacer grupos pequeños de trabajo sobre todo con las tareas de vistas para mejorar la eficiencia, también tenemos que investigar una herramienta para la creación de vistas ya que es algo que resulta demasiado tedioso para los resultados que se obtienen, si no encontramos ninguna debemos hacer componentes modulares para crear nuevas vistas usándolos.

10.4. Conclusión de la reunión

La reunión tuvo una duración de *45 minutos*, participó todo el equipo *Scrum* de forma dinámica, se siguió un modelo de debate libre en el que respetamos los turnos de palabra y se hablaba por orden de petición, las peticiones se hacían alzando la mano.