PLANNING SPRINT

Juan José Guerrero Ramirez

Mikey Collazos

Brayan Turmeque Castiblanco

Universidad Manuela Beltran

ARQUITECTURA DEL SOFTWARE\_A2

9/03/2025

# Definition of Done

We decided to have an online meeting to define what is going to be our "Definition of Done" for the development of the Virtual Board Games and Collectible Cards Store.

We considered different perspectives within the team and agreed on a set of standards to determine when a feature is fully completed.

To define this, we focused on documentation, structured code, functionality validation, and testing processes. Additionally, we used Planning Poker and WSJF analysis to evaluate the complexity of each user story and prioritize the most challenging ones at the beginning of the development process.

### Definition of Done Criteria

1. All User Stories must be fully implemented, ensuring they pass the acceptance criteria checks established in the backlog.
2. A User Story is considered completed only when all acceptance criteria have been validated and tested under real scenarios. No exceptions, every requirement must be met.
3. Sprints must deliver tested and validated code, ensuring that every module works as expected before moving to the next sprint.
4. At this stage, we are not yet deploying to a live production environment, but all functionalities must be fully functional and tested in a staging environment.
5. We can proceed to a new sprint only if at least three critical functionalities have been successfully validated in the current sprint.
6. Given the varying difficulty of different User Stories, we prioritize the most complex and essential features at the beginning of the development. If we successfully complete the hardest tasks early on, the rest of the system will be easier to build and integrate.

**HISTORIAS DE USUARIO DETALLADAS**

## Registro de Usuario (User Registration) **Prioridad:** Alta **Puntos de Esfuerzo:**

### Descripción

*Como nuevo usuario, quiero registrarme en la plataforma proporcionando mi información personal para poder acceder a mi cuenta y realizar compras de manera segura.*

### Criterios de Aceptación

1. El sistema debe permitir ingresar **nombre, correo electrónico y contraseña**.
2. La contraseña debe seguir reglas de seguridad (**mínimo 8 caracteres, una mayúscula y un número**).
3. No se debe permitir registrar una cuenta con un **correo electrónico existente**.
4. Se debe enviar un **correo de confirmación** al usuario después del registro.
5. El usuario no podrá **iniciar sesión hasta confirmar su cuenta**.

### Tareas de Desarrollo

* Implementar un **formulario de registro con validaciones**.
* Diseñar y desarrollar una **API de registro en Spring Boot**.
* Almacenar datos del usuario en una **base de datos MySQL**.
* Implementar seguridad en la contraseña utilizando **BCrypt**.

## Inicio de Sesión (User Login) **Prioridad:** Alta **Puntos de Esfuerzo:**

### Descripción

*Como usuario registrado, quiero iniciar sesión en la plataforma usando mi correo y contraseña para acceder a mi cuenta y realizar compras.*

### Criterios de Aceptación

1. El usuario debe ingresar un **correo válido y una contraseña**.
2. Si las credenciales son incorrectas, el sistema debe mostrar un **mensaje de error**.
3. La autenticación debe manejarse con **JWT** para mayor seguridad.
4. Se debe ofrecer la opción **"Recuérdame"** con una **cookie segura**.
5. Un usuario con una cuenta **no confirmada** no podrá iniciar sesión.

### Tareas de Desarrollo

* Implementar un **formulario de inicio de sesión con validaciones**.
* Diseñar una **API de autenticación con JWT en Spring Boot**.
* Gestionar sesiones activas con **cookies seguras y tokens**.
* Validar que solo usuarios **confirmados** puedan iniciar sesión.
* Proteger rutas privadas con **autenticación**.

## Recuperación de Contraseña (Password Recovery)

**Prioridad:** Media  
 **Puntos de Esfuerzo:**

### Descripción

*Como usuario registrado, quiero recuperar mi contraseña si la olvido para poder acceder nuevamente a mi cuenta.*

### Criterios de Aceptación

1. El sistema debe permitir a los usuarios ingresar su **correo para recibir un enlace de recuperación**.
2. Si el correo no está registrado, se debe mostrar un **mensaje de error**.
3. Se debe enviar un **enlace seguro** para restablecer la contraseña.
4. La nueva contraseña debe seguir **las reglas de seguridad**.
5. Una vez restablecida, el usuario debe poder **iniciar sesión normalmente**.

### Tareas de Desarrollo

* Implementar un **formulario de recuperación de contraseña**.
* Crear una **API para enviar enlaces de restablecimiento por correo**.
* Validar reglas de seguridad para la nueva contraseña.
* Implementar **expiración del enlace de recuperación** por seguridad.

## 4 Visualización del Catálogo de Productos (Product Catalog View) **Prioridad:** Alta **Puntos de Esfuerzo:**

### Descripción

*Como usuario, quiero navegar por un catálogo de juegos de mesa y cartas coleccionables para explorar productos y elegir qué comprar.*

### Criterios de Aceptación

1. El catálogo debe mostrar **imágenes, nombres, descripciones y precios** de los productos.
2. Los productos deben estar organizados **por categorías**.
3. Se debe incluir **barra de búsqueda y filtros** para encontrar productos rápidamente.
4. La tienda debe ser **rápida y responsiva** en computadoras y móviles.
5. Al hacer clic en un producto, se debe mostrar una **descripción detallada**.

### Tareas de Desarrollo

* Crear una **base de datos MySQL** para almacenar productos.
* Desarrollar una **API REST para listar productos**.
* Implementar la interfaz con **React o Vue**.
* Agregar **filtros dinámicos y barra de búsqueda**.
* Optimizar las imágenes para **carga rápida**.

## Carrito de Compras (Shopping Cart) **Prioridad:** Alta **Puntos de Esfuerzo:** **1**

### Descripción

*Como usuario, quiero agregar productos a un carrito de compras para organizarlos antes de pagar.*

### Criterios de Aceptación

1. Los usuarios deben poder **guardar productos en el carrito** hasta que sean eliminados o comprados.
2. Se debe permitir **modificar la cantidad** de productos en el carrito.
3. Los usuarios deben poder **eliminar productos fácilmente**.
4. El **total de la compra** debe actualizarse en tiempo real.
5. Si el usuario cierra sesión y vuelve a ingresar, los productos deben **mantenerse en el carrito**.

### Tareas de Desarrollo

* Crear una **API REST para gestionar el carrito**.
* Almacenar el carrito en **base de datos y cookies**.
* Implementar la **interfaz con actualizaciones en tiempo real**.
* Validar **disponibilidad del producto** antes de agregarlo al carrito.
* Agregar un **botón para vaciar el carrito** con un clic.

# 

# Guiding Questions

It refers to the structures of a system composed of elements with externally visible properties and

relations between them. This definition is:

1. UML

2. Software Architecture

3. View

4. Controller

It is a technical leader must know the principles related to software architecture, have extensive

knowledge about technology and have excellent skills in written and oral communication. The previous

profile refers to:

1. developer

2. Software architect

3. Tester

4. Analys

# 

# Prioritized backlog

MoscoW

| Historia de Usuario | Clasificación MoSCoW |
| --- | --- |
| User Registration | Must Have |
| User Login | Must Have |
| Payment Processing | Must Have |
| Product Catalog View | Should Have |
| Password Recovery | Should Have |
| Shopping Cart | Should Have |

# WSJF

| Historia de usuario | Valor de negocio | Urgencia | Riesgo-oportunidad | Esfuerzo | WSJF |
| --- | --- | --- | --- | --- | --- |
| User Registration | 5 | 5 | 5 | 5 | 3.0 |
| User Login | 5 | 5 | 4 | 2 | 7.0 |
| Product Catalog View | 5 | 5 | 5 | 5 | 3.0 |
| Payment Processing | 5 | 5 | 5 | 8 | 1.88 |
| Shopping Cart | 3 | 3 | 2 | 3 | 2.67 |
| Password Recovery | 3 | 3 | 2 | 5 | 1.6 |

# 

# Planning Poker

| **User Registration** |
| --- |
|  |

| **Historia 2- User login** |
| --- |
|  |

| **Historia 3- Password Recovery** |
| --- |
|  |

| **Historia 4- Product Catalog View** |
| --- |
|  |

| **Historia 5-Shopping Cart** |
| --- |
|  |

| **Historia 6- Payment Preocessing** |
| --- |
|  |

| **Estrategias** | | **Encargado** |
| --- | --- | --- |
| **Aspectos técnicos** | Juan, como Líder Técnico, garantiza una arquitectura eficiente y un desarrollo sólido en Java, JSP y Servlets. Colabora estrechamente con el equipo para alinear la implementación con los plazos y la calidad del proyecto. Su enfoque en resolver problemas técnicos, optimizar el código y coordinar con el líder organizacional y de calidad asegura un software funcional, escalable y bien estructurado. | **Juan** |
| **Comunicación** | como líder de organización y comunicación, garantiza un flujo de trabajo estructurado y una experiencia de usuario fluida, asegurando la coherencia en la navegación, carga de datos y gestión de URLs. Su enfoque en la optimización y prevención de errores, junto con su comunicación efectiva, mantiene alineado al equipo. Para maximizar su impacto, puede apoyarse en herramientas de gestión, reuniones periódicas y una estrecha colaboración con el líder técnico. | **Brayan** |
| **Aseguramiento de**  **calidad** | Mikey, como líder de aseguramiento de calidad, garantiza un software estable y libre de errores mediante pruebas rigurosas y validación de requisitos. Su enfoque en la detección temprana de fallos y optimización del rendimiento asegura una experiencia de usuario fluida y confiable. Además, colabora estrechamente con el equipo técnico y organizacional para alinear los estándares de calidad con los objetivos del proyecto. Para maximizar su impacto, puede apoyarse en pruebas automatizadas, revisión continua y herramientas de gestión de calidad. | **Mikey** |

# Work Schedule

## Plan created in Jira:

