

Documento de Análisis y Diseño del Juego Terapéutico

1. Fase de Requerimientos:

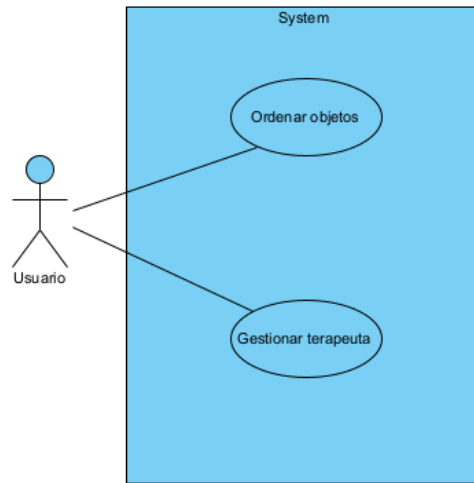


Figura 1. Diagrama de casos de uso

2. Fase de Análisis

a. Caso de uso: Ordenar objetos

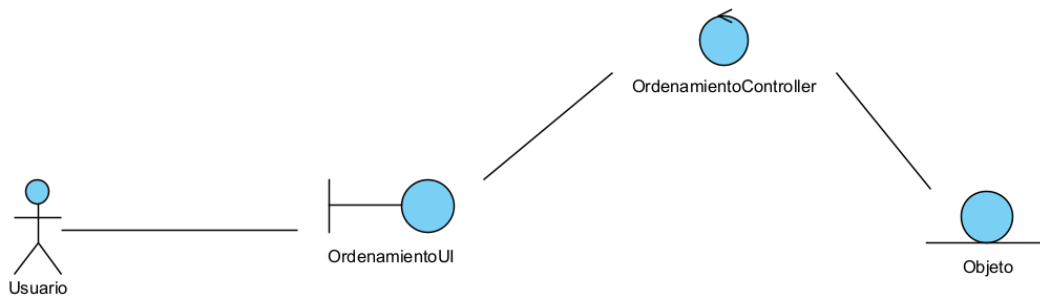


Figura 2. Diagrama de robustez del primer caso de uso

b. Caso de uso: Gestionar terapeuta

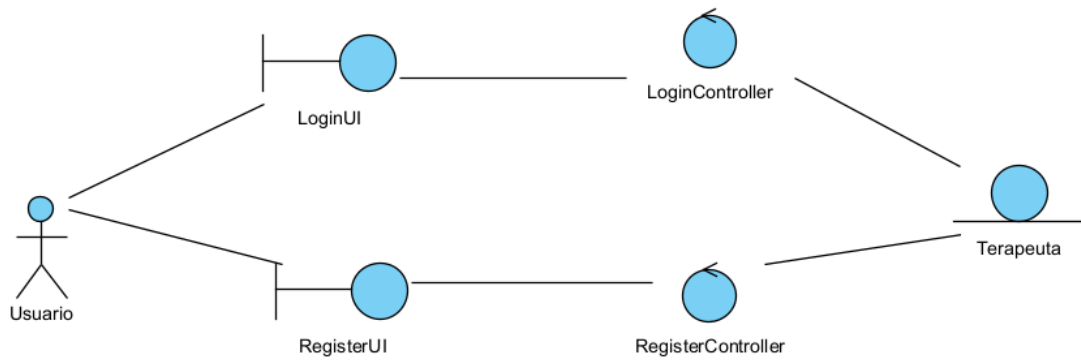


Figura 3. Diagrama de robustez del segundo caso de uso

3. Fase de Diseño

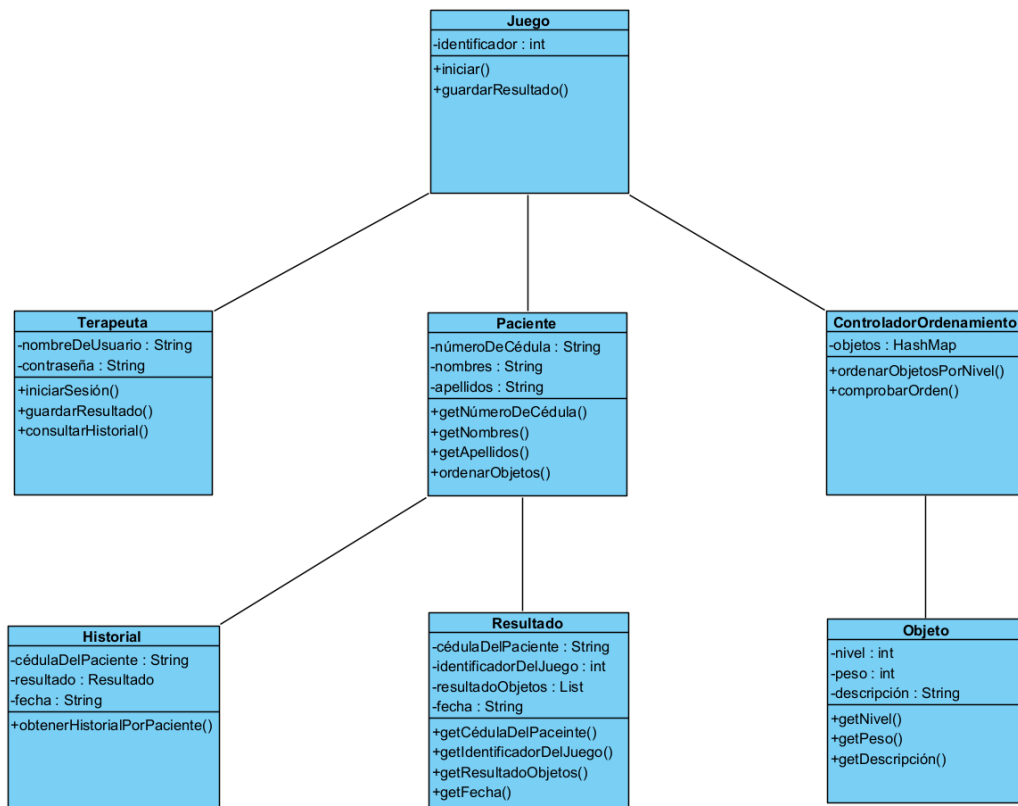


Figura 4. Diagrama de clases

4. Especificación de la Arquitectura

Se desarrollará un aplicativo web, el cual está estructurado en una arquitectura Modelo-Vista-Controlador (MVC). A continuación, se detallan los componentes principales y cómo se integran entre sí:

- **Modelo:**

- **Hibernate** se emplea como el ORM (Object-Relational Mapping) para gestionar la persistencia de los datos en la base de datos.
- Las clases de entidad de Hibernate representan los datos y permiten realizar transacciones seguras, asegurando que la información sobre el progreso de los usuarios, los objetos, y el historial sea almacenada y gestionada eficientemente.

- **Vista:**

- Se utilizará **Java Server Pages (JSP)** para crear las interfaces de usuario dinámicas que permiten la interacción directa de los usuarios con el sistema.
- Cada página JSP estará organizada para garantizar accesibilidad y usabilidad, empleando un diseño intuitivo para que los usuarios interactúen fácilmente con el juego, ordenen objetos y revisen su historial.

- **Controlador:**

- Los **Servlets** se utilizan para manejar las solicitudes HTTP y coordinar las operaciones entre la vista y el modelo.
- Los controladores gestionan la lógica de negocio, validan la entrada de datos del usuario y, en función del caso de uso (por ejemplo, "Ordenar objetos" o "Ver historial"), invocan los métodos del modelo que corresponden a la acción solicitada, asegurando una respuesta rápida y segura.

- **Gestión de Dependencias:**

- **Maven:** Con Maven se organiza y gestiona el ciclo de vida del proyecto, lo que facilita la integración de bibliotecas externas y frameworks como Hibernate y JSP.

- **Flujo de la Aplicación:**

- El flujo general del aplicativo empieza con una solicitud del usuario (a través de la interfaz JSP) que se envía al servlet correspondiente. El servlet procesa la solicitud, invoca las operaciones necesarias en el modelo, y envía la información actualizada de vuelta a la vista. Además, en caso de ser necesario, se guardará información en la base de datos. Este ciclo asegura que todas las interacciones sean procesadas en tiempo real y reflejadas en la interfaz de usuario.

Listado de Herramientas y Tecnologías de Desarrollo:

- **Control de versiones:** GitHub
- **IDE:** VsCode, IntelliJ IDEA
- **Herramienta CASE:** Visual Studio Paradigm
- **Gestión de dependencias:** Maven
- **Desarrollo Web:** JSP
- **Servidor Web:** Tomcat 9
- **Bases de datos:** pgAdmin, PostgreSQL
- **ORM:** Hibernate