



PRD-VER-COD

Modelos de Clases, Objetos y Diagrama de Estados.

Chat Web Grupal - Chatly

Versión 1.0.0

Área de Desarrollo
Dirección de Tecnología

CONFIDENCIAL

Tabla de Contenido

1. Introducción	2
1.1. Definiciones, acrónimos y abreviaturas	2
2. Antecedentes	2
3. Alcance	2
4. Componentes del sistema	3
5. Modelo de Clases	3
6. Modelo de Objetos	6
6.1. Modelo General de objetos	6
6.2. Modelos específicos de objetos	7
6.2.1. Modelo de Objeto 1: Usuario y Preferencias	7
6.2.2. Modelo de Objeto 2: ChatRoom y Mensajes	7
6.2.3. Modelo de Objeto 3: Usuario Enviando un Mensaje	8
7. Diagrama de Estados	9
8. Firma de Participantes	12



Modelos de Clases, Objetos y Diagrama de Estados

Chat Web Grupal

1. Introducción

Este documento detalla el diseño y los modelos del sistema Chatly, incluyendo el modelo de clases, objetos y estados, que definen la estructura lógica del sistema y su comportamiento.

Chatly es un sistema web para comunicación en tiempo real que facilita la interacción grupal mediante salas de chat, autenticación segura y notificaciones instantáneas.

1.1. Definiciones, acrónimos y abreviaturas


Término	Descripción
JWT	JSON Web Token.
WebSocket	Protocolo para comunicación en tiempo real.
ChatRoom	Espacio virtual para mensajes grupales.
API	Interfaz de programación de aplicaciones.
Interfaz	Es la parte del software que interactúa con el usuario, o a interfaces de programación, que definen cómo los diferentes componentes del software deben comunicarse.

2. Antecedentes

El desarrollo de un sistema de chat en tiempo real responde a la necesidad de mejorar la comunicación entre usuarios en entornos digitales mediante el uso de tecnologías modernas como WebSockets, Angular y Spring Boot. Actualmente, las soluciones existentes carecen de una integración robusta que permita manejar usuarios concurrentes, garantizar la seguridad de datos y ofrecer notificaciones en tiempo real.

3. Alcance

El sistema permitirá una comunicación en tiempo real entre múltiples usuarios, la gestión de salas de chat grupales, autenticación segura mediante tokens JWT y la Visualización de usuarios

	CHATLY Chat Web Grupal	
	Modelos de Clases, Objetos y Diagrama de Estados.	Versión: 1.0
		Página 3 de 13

conectados y notificaciones instantáneas. No incluye: Integración con redes sociales o herramientas de videoconferencia.

4. Componentes del sistema

Módulo de Gestión de Usuarios: Registro, inicio de sesión y gestión de sesiones.

Módulo de Salas de Chat: Creación y administración de salas grupales (Crear, Unirse y Abandonar).

5. Modelo de Clases

El modelo de clases del sistema Chatly define las entidades principales y sus interacciones, proporcionando una estructura detallada del sistema de chat web grupal. Este modelo incluye las clases User, UserPreference, ChatRoom, ChatMessage, y los servicios asociados para la autenticación y gestión de salas de chat.

- **Clase User**
Descripción: Representa a los usuarios del sistema, almacenando información básica para su identificación y autenticación.
- **Atributos:**
 - id: Identificador único del usuario (tipo: bigint).
 - username: Nombre de usuario único (tipo: String).
 - password: Contraseña del usuario (tipo: String).
 - role: Rol del usuario en el sistema (tipo: String). Puede ser USER o ADMIN.
 - created_at: Fecha de creación de la cuenta (tipo: Timestamp).
 - user_preferences_id: Referencia a las preferencias del usuario (tipo: bigint).
- **Métodos:**
 - login(): Autentica al usuario mediante credenciales.
 - logout(): Cierra la sesión del usuario.
- **Clase UserPreference**
- **Descripción:** Almacena las preferencias personalizadas de un usuario, como el tema visual y el estado.
- **Atributos:**
 - id: Identificador único de las preferencias (tipo: bigint).
 - theme: Tema visual preferido por el usuario (tipo: String).
 - color: Color principal seleccionado (tipo: String).
 - avatar: URL de la imagen de perfil del usuario (tipo: String).
 - about: Descripción del usuario (tipo: String).
 - status: Estado actual del usuario (ONLINE, OFFLINE, etc.) (tipo: String).
- **Clase ChatRoom**
- **Descripción:** Representa una sala de chat grupal donde los usuarios pueden enviar y recibir mensajes.
- **Atributos:**
 - id: Identificador único de la sala de chat (tipo: bigint).
 - name: Nombre de la sala de chat (tipo: String).



- description: Descripción de la sala (tipo: String).
 - created_at: Fecha y hora de creación (tipo: Timestamp).
- **Métodos:**
 - createRoom(name: String, description: String): Crea una nueva sala de chat.
 - addParticipant(user: User): Agrega un usuario a la sala.
 - removeParticipant(user: User): Elimina a un usuario de la sala.
- **Clase ChatMessage**
- **Descripción:** Representa un mensaje enviado por un usuario en una sala de chat.
- **Atributos:**
 - id: Identificador único del mensaje (tipo: bigint).
 - chat_room_id: Referencia a la sala de chat asociada (tipo: bigint).
 - user_id: Referencia al usuario que envió el mensaje (tipo: bigint).
 - message: Contenido del mensaje (tipo: String).
 - created_at: Fecha y hora de creación del mensaje (tipo: Timestamp).
 - edited: Indica si el mensaje fue editado (tipo: boolean).
 - deleted: Indica si el mensaje fue eliminado (tipo: boolean).
- **Métodos:**
 - sendMessage(chatRoomId: bigint, message: String): Envía un mensaje a la sala.
 - editMessage(newContent: String): Edita el contenido de un mensaje existente.
 - deleteMessage(): Marca el mensaje como eliminado.
- **Clase AuthenticationService**
- **Descripción:** Proporciona operaciones relacionadas con la autenticación y la gestión de usuarios.
- **Métodos:**
 - authenticate(username: String, password: String): Boolean: Verifica las credenciales del usuario.
 - generateToken(user: User): String: Genera un token JWT para la autenticación.
 - logout(user: User): Finaliza la sesión del usuario.
- **Clase ChatRoomService**
- **Descripción:** Gestiona la lógica de las salas de chat y las interacciones entre usuarios y mensajes.
- **Métodos:**
 - createChatRoom(name: String, description: String): ChatRoom: Crea una sala nueva.
 - addParticipant(chatRoom: ChatRoom, user: User): Agrega un participante a la sala.
 - removeParticipant(chatRoom: ChatRoom, user: User): Elimina un participante de la sala.
 - getMessages(chatRoomId: bigint): List<ChatMessage>: Obtiene todos los mensajes de una sala específica.

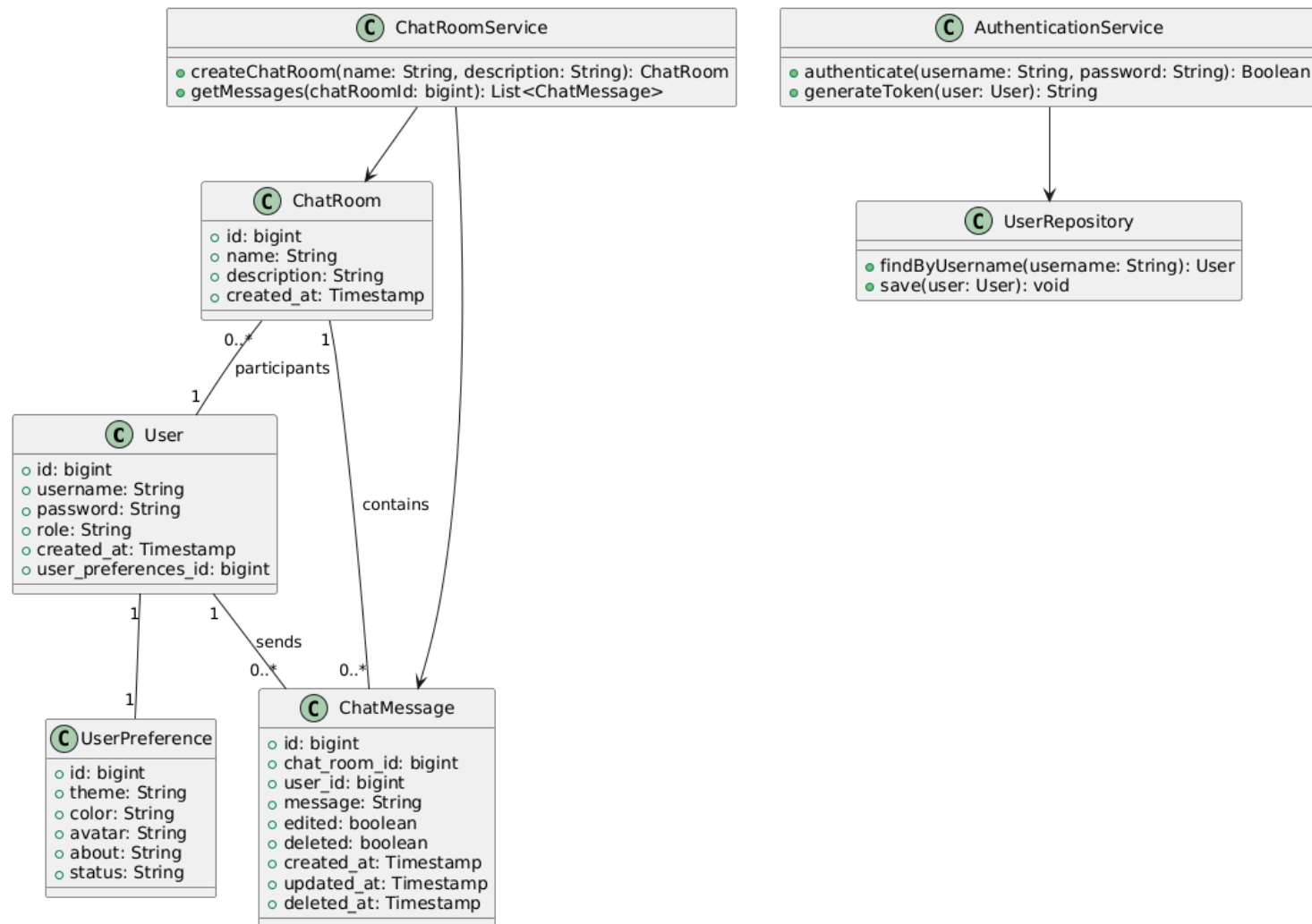


CHATLY Chat Web Grupal

Modelos de Clases, Objetos y Diagrama de Estados.

Versión: 1.0

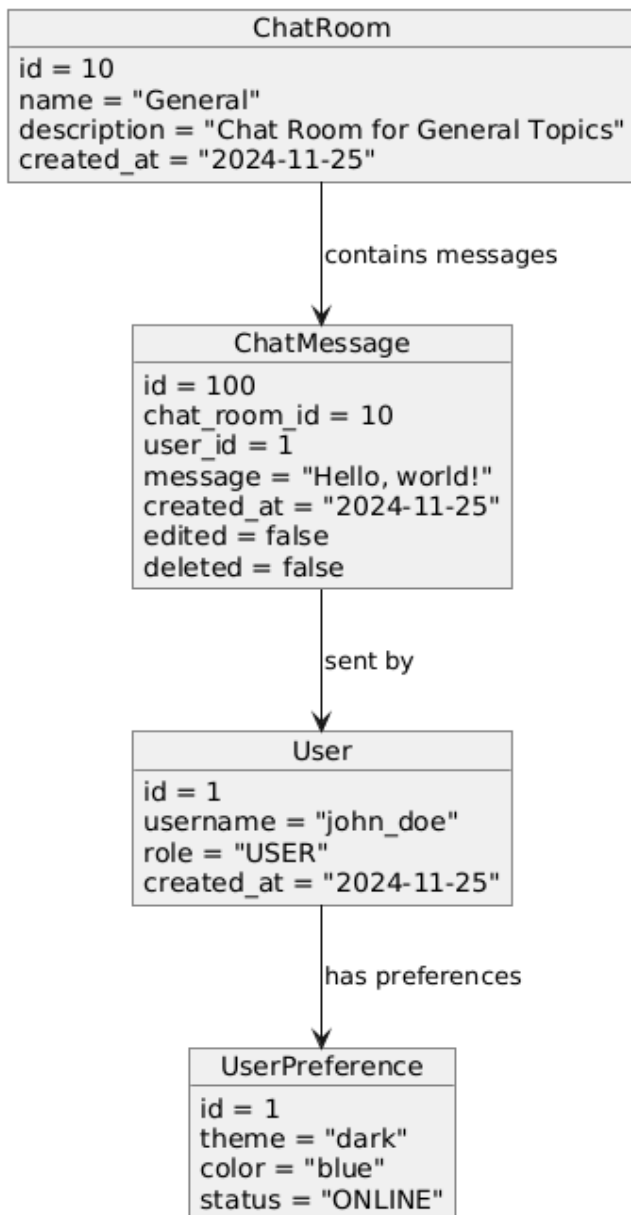
Página 5 de 13





6. Modelo de Objetos

6.1. Modelo General de objetos



El modelo de objetos proporciona una representación en tiempo de ejecución de las clases y su interacción dentro del sistema Chatly. A continuación, se describe un modelo general de objetos y tres modelos específicos que representan escenarios clave del sistema.

El modelo general de objetos describe cómo las instancias de las clases principales (User, ChatRoom, ChatMessage, UserPreference) interactúan entre sí en un momento específico de la ejecución del sistema.

Descripción:

- Una instancia de **User** está asociada a una instancia de **UserPreference**, que contiene la configuración personalizada del usuario.
- Múltiples instancias de **User** pueden estar asociadas a una misma instancia de **ChatRoom**.
- Cada instancia de **ChatRoom** contiene múltiples instancias de **ChatMessage**.
- Las instancias de **ChatMessage** están vinculadas a las de **User** para identificar al remitente.



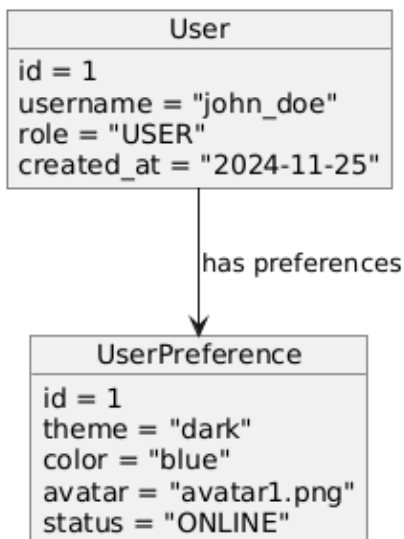
6.2. Modelos específicos de objetos

Los modelos específicos representan casos de uso concretos y cómo los objetos interactúan en esos escenarios.

6.2.1. Modelo de Objeto 1: Usuario y Preferencias

Descripción: Este modelo muestra cómo una instancia de User está relacionada con su configuración personalizada, almacenada en un objeto de tipo UserPreference.

Escenario: El usuario John Doe tiene configuradas sus preferencias:



- Tema: Oscuro.
- Color: Azul.
- Avatar: john_avatar.png.
- Estado: Online.

Diagrama:

- Objeto User:
 - id = 1.
 - username = "john_doe".
- Objeto UserPreference:
 - theme = "dark".
 - color = "blue".
 - avatar = "john_avatar.png".
 - status = "ONLINE".

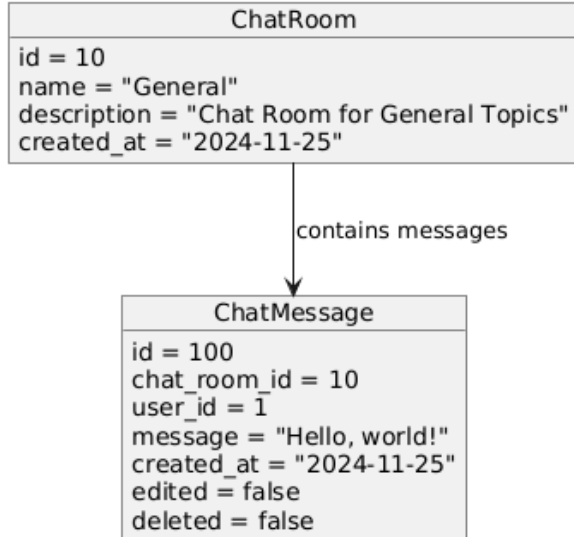
6.2.2. Modelo de Objeto 2: ChatRoom y Mensajes

Descripción: Este modelo describe la interacción entre una sala de chat (ChatRoom) y los mensajes (ChatMessage) enviados por los usuarios.

Escenario: La sala de chat General contiene múltiples mensajes enviados por varios usuarios. Uno de los mensajes es de John Doe, quien escribió: "Hola, equipo".



Diagrama:



- **Objeto ChatRoom:**
 - id = 10.
 - name = "General".
 - description = "Sala de chat general".
- **Objeto ChatMessage:**
 - id = 100.
 - chatRoomId = 10.
 - userId = 1.
 - message = "Hola, equipo".
 - createdAt = "2024-11-25 10:00:00".

6.2.3. Modelo de Objeto 3: Usuario Enviando un Mensaje

Descripción: Este modelo muestra el proceso de envío de un mensaje por parte de un usuario a una sala de chat.

Escenario: El usuario John Doe envía el mensaje: "¿Cómo están?" a la sala de chat General.

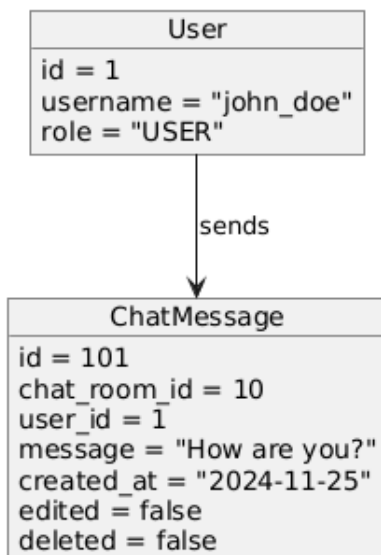


Diagrama:

- **Objeto User:**
 - id = 1.
 - username = "john_doe".
- **Objeto ChatRoom:**
 - id = 10.
 - name = "General".
- **Objeto ChatMessage:**
 - id = 101.
 - chatRoomId = 10.
 - userId = 1.
 - message = "¿Cómo están?".
 - createdAt = "2024-11-25 10:05:00".



7. Diagrama de Estados

El diagrama de estados ilustra el ciclo de vida de la sesión de un usuario en el sistema Chatly, desde el momento en que intenta iniciar sesión hasta que cierra su sesión. A continuación, se describen los estados y las transiciones representadas en el diagrama.

- **Estados**

- **Estado Inicial (●):**

- Punto de inicio del flujo, donde el sistema está listo para recibir una acción del usuario.

- **Login:**

- El usuario ingresa sus credenciales (nombre de usuario y contraseña).
 - Es el estado donde se captura la información necesaria para autenticar al usuario.

- **Authenticating:**

- El sistema verifica las credenciales ingresadas llamando al repositorio de usuarios (UserRepository).
 - Durante este estado, se validan el nombre de usuario y la contraseña contra la base de datos.

- **Error:**

- Estado alcanzado si las credenciales ingresadas son inválidas.
 - El sistema muestra un mensaje de error al usuario y le permite intentar nuevamente.

- **Success:**

- Estado alcanzado cuando las credenciales ingresadas son válidas.
 - Se genera un token de autenticación (JWT) para la sesión del usuario.
 - El sistema redirige al usuario al menú principal (MainMenu).

- **MainMenu:**

- Estado donde el usuario interactúa con las opciones del sistema, como unirse a salas de chat o enviar mensajes.
 - Es un estado intermedio que representa la funcionalidad principal del sistema.

- **Logout:**

- Estado final al que se llega cuando el usuario selecciona la opción de cerrar sesión.
 - Se destruye el token de autenticación y se finaliza la sesión.

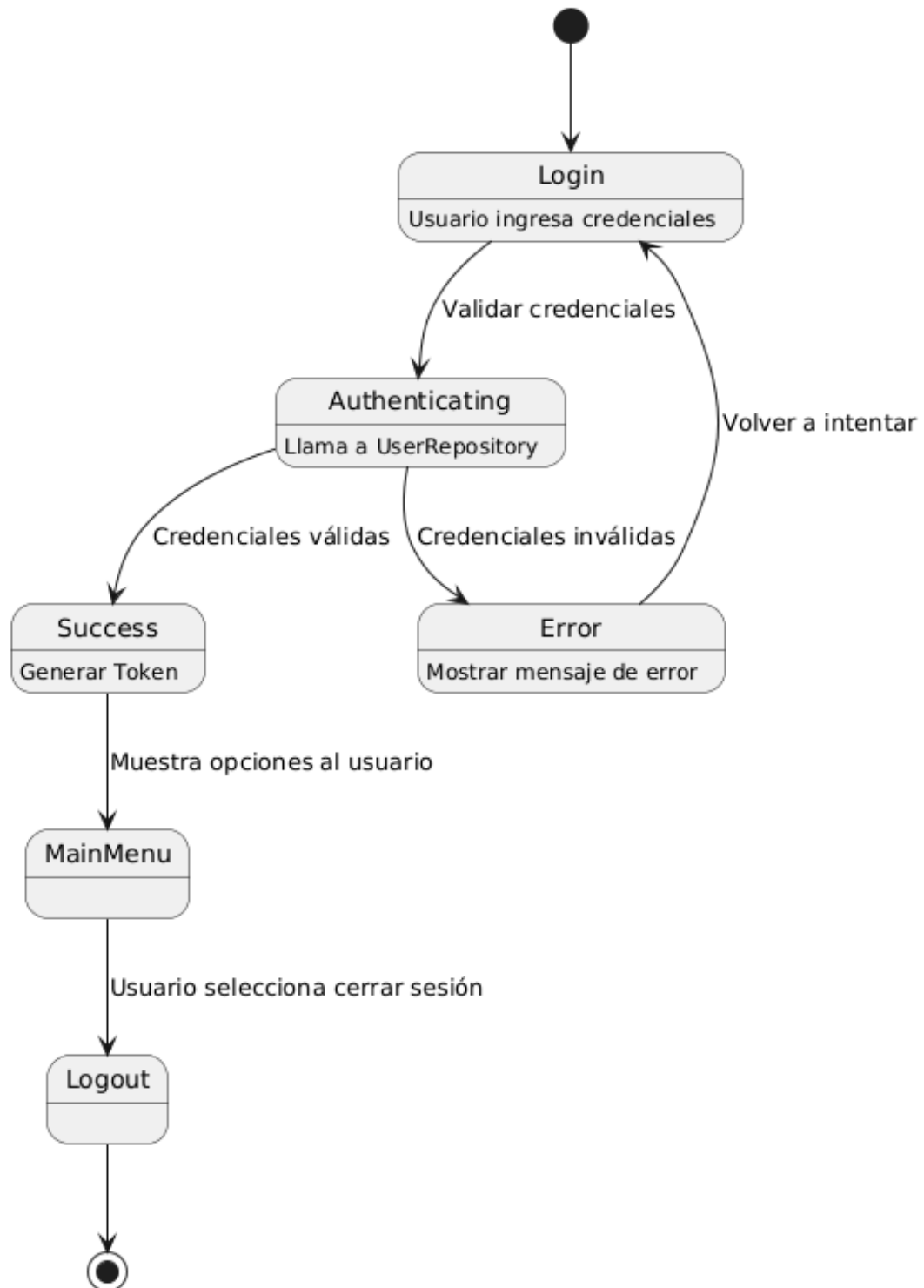
- **Estado Final (●):**

- Punto donde el flujo de interacción del usuario con el sistema termina.

- **Transiciones**



- De Estado Inicial a Login:
 - Ocurre cuando el sistema se carga y está listo para aceptar las credenciales del usuario.
- De Login a Authenticating:
 - Transición activada cuando el usuario ingresa credenciales y el sistema comienza la validación.
- De Authenticating a Error:
 - Se produce si las credenciales ingresadas son inválidas.
 - El sistema redirige al usuario para que intente ingresar nuevamente.
- De Error a Login:
 - El usuario decide intentar ingresar nuevas credenciales después de un intento fallido.
- De Authenticating a Success:
 - Transición activada cuando el sistema valida con éxito las credenciales del usuario.
- De Success a MainMenu:
 - El sistema redirige al menú principal después de generar un token de autenticación válido.
- De MainMenu a Logout:
 - Se produce cuando el usuario selecciona la opción de cerrar sesión.
- De Logout a Estado Final:
 - Marca el final del flujo de la sesión del usuario.





8. Firma de Participantes

APROBACIÓN DEL CRONOGRAMA

Elaborado y Aprobado		
Unidad de Negocio	Responsable	Firma
Equipo de Desarrollo	Grupo 11	
Docente	Ing. Ángel Marcelo Rea Guamán.	
FECHA DE APROBACIÓN: 25/11/2024		