

Introducción

Introducción al proyecto

Propósito

Objetivos

Fase de Análisis

En la fase de análisis se pretende obtener un conocimiento razonable del sistema. Comenzaremos definiendo los requerimientos funcionales y no funcionales del sistema, después describiremos los casos de uso y por último definiremos el modelo del dominio.

(introducción)

Requerimientos

Funcionales:

❖ Acciones del reproductor:

- **Añadir Canción/Lista de Reproducción:** El usuario podrá añadir las canciones que desee, usando el enlace de esta, o utilizando el enlace a una lista de reproducción, el usuario podrá insertar un enlace incorrecto.
- **Borrar Canción/Lista de Reproducción:** El usuario podrá eliminar canciones ya añadidas a la lista de canciones de la aplicación.
- **Subir/Bajar Volumen:** El usuario podrá cambiar el volumen de su reproductor utilizando un “slider”, la cantidad exacta de volumen no es fundamental.
- **Omitir Canción:** El usuario podrá omitir la canción en reproducción en ese momento, siempre que el propio usuario haya solicitado su reproducción, o una mayoría de usuarios en la sala voten el omitir la misma.

- ❖ **Iniciar Sesión:** El usuario podrá identificarse en la aplicación, deberá introducir sus datos de identificación, estos podrán ser incorrectos, en caso contrario, el usuario será redireccionado a la página inicial habiéndose identificado.

- ❖ **Registrarse:** El usuario podrá registrarse en la aplicación, deberá introducir sus datos de identificación, estos podrán ser incorrectos, en caso contrario, el usuario será redireccionado a la página inicial habiéndose identificado.
- ❖ **Crear Sala:** El usuario podrá crear una sala desde la cual podrá realizar el resto de las acciones del reproductor.
- ❖ **Unirse a Sala:** El usuario podrá unirse a una sala ya creada, para ello el usuario deberá conocer el identificador de la sala a la que desea unirse, desde la sala el usuario podrá realizar el resto de las acciones del reproductor.

No Funcionales:

- ❖ **Rendimiento:** el servidor debe de servir los datos necesarios en un tiempo aceptable y constante.
- ❖ **Disponibilidad:** el servicio debe de estar disponible las 24/7, siempre que se disponga de una conexión a internet y un equipo desde el que acceder
- ❖ **Estabilidad:** El servicio debe ser estable y evitar las caídas de servicio en la medida de lo posible
- ❖ **Accesibilidad:** la aplicación debe ser accesible para usuarios familiarizados con el uso de este tipo de servicios, como para usuarios sin tanta experiencia en el uso de estas tecnologías, debe ser visualmente sencilla de comprender.
- ❖ **Documentación:** la aplicación contará con un apartado en el que se muestre cómo usar la misma.
- ❖ **Mantenibilidad:** la aplicación debe ser fácilmente mantenible, una de las prioridades será la escalabilidad de la aplicación, por consecuencia facilitando el mantenimiento de esta.
- ❖ **Escalabilidad:** desde el primer momento en el desarrollo será necesario tener en cuenta la posible adición de nuevas características.
- ❖ **Seguridad:** la aplicación debe de ser segura, se filtrarán los datos ofrecidos por el usuario y se podrá hacer uso de la una vez el usuario se haya registrado y sus datos queden guardados en el sistema.
- ❖ **Privacidad:** Los datos de los usuarios registrados serán completamente privados y ningún otro usuario tendrá acceso a los datos del resto de usuarios.

Casos de Uso

Diagrama de frontera de Sirecope:

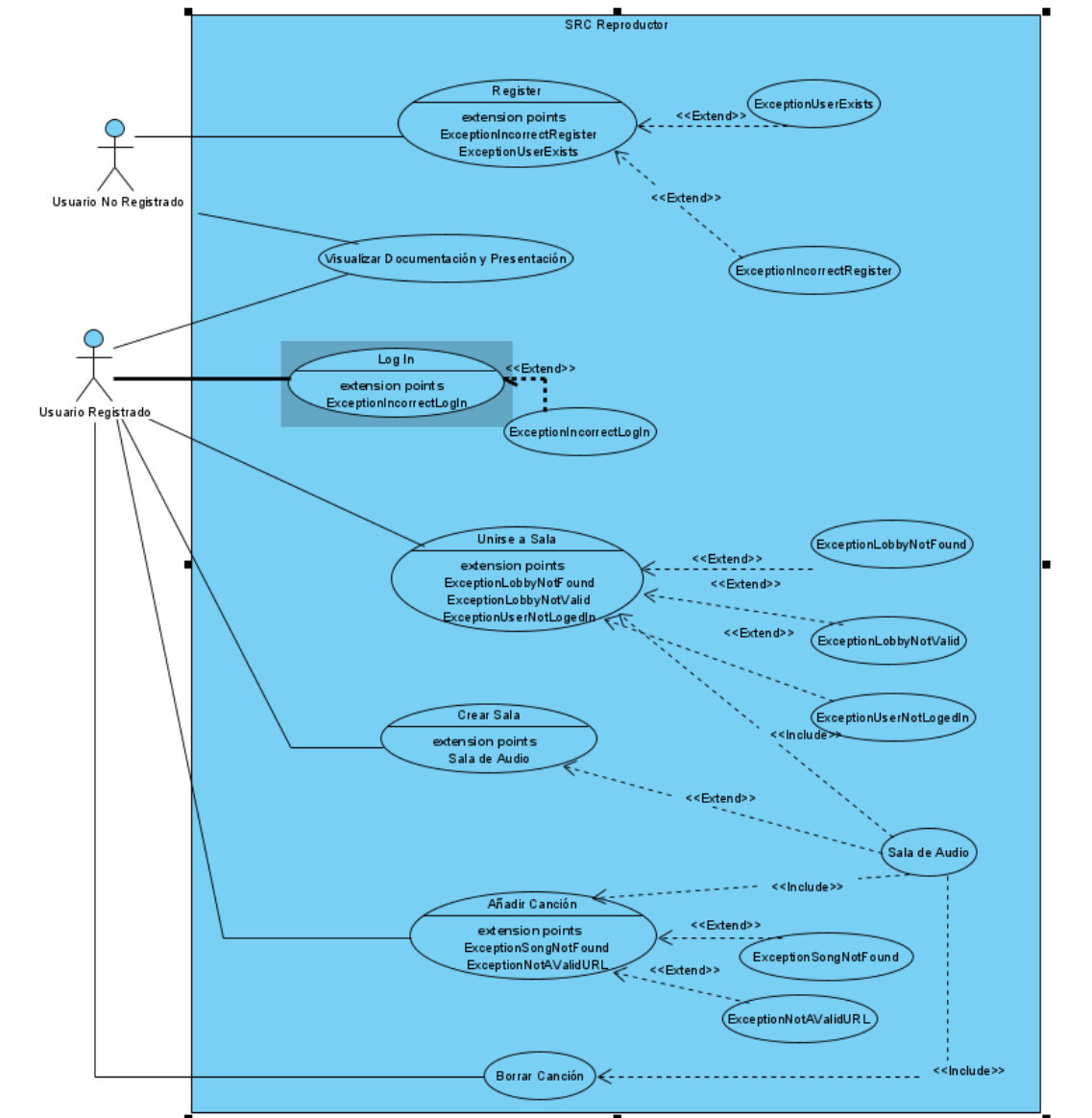
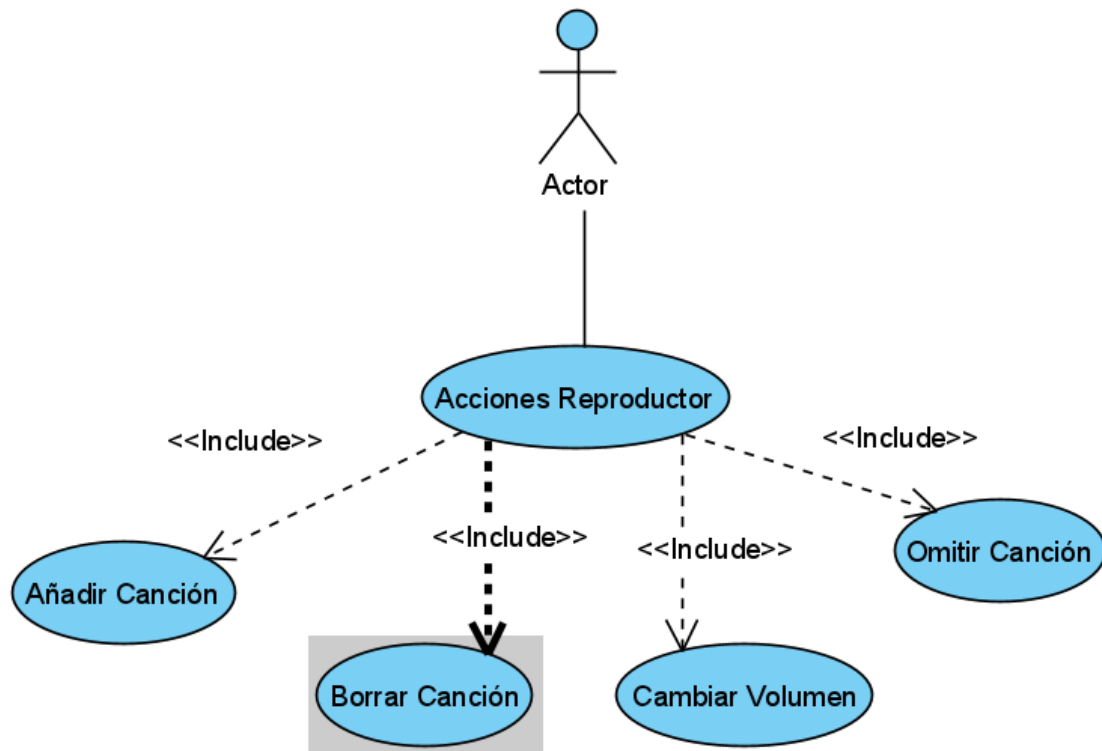


Diagrama Casos de Uso Reproductor.



Caso de Uso "Register"

- Actor: Usuario no registrado
- Condiciones de entrada: El usuario no se encuentra registrado en la aplicación
- Condiciones de salida: El usuario registra sus datos y puede acceder con los mismos al sistema
- Flujo de eventos:
 - ❖ El usuario decide registrarse
 - ❖ La aplicación solicita la información necesaria
 - ❖ El usuario proporciona la información
 - ❖ El sistema verifica la información y, en caso de que reconozca como válidos los datos, se informa al usuario de el registro correcto.
- Excepciones:
 - ExceptionUserExists: Los datos proporcionados ya existen.
 1. La aplicación muestra mensaje de error "El usuario ya existe"
 2. Vuelve a solicitar los datos del usuario
 - ExceptionIncorrectRegister: Los datos proporcionados no son válidos.

1. La aplicación muestra mensaje de error “Los datos no son válidos”
2. Vuelve a solicitar los datos del usuario

Caso de Uso “Log In”

- Actor: Usuario Registrado
- Condiciones de entrada: El usuario se encuentra registrado en la aplicación
- Condiciones de salida: El usuario se identifica en la aplicación
- Flujo de eventos:
 - ❖ El usuario decide Identificarse
 - ❖ La aplicación solicita la información necesaria
 - ❖ El usuario proporciona la información
 - ❖ El sistema verifica la información y, en caso de que reconozca como válidos los datos, se informa al usuario de la correcta identificación.
- Excepciones:
 - ExceptionIncorrectLogin: Los datos proporcionados no corresponden a ningún usuario.
 1. La aplicación muestra mensaje de error “El usuario no existe o los datos son incorrectos”
 2. Vuelve a solicitar los datos del usuario

Caso de Uso “Visualizar Documentación y Presentación”

- Actor: Usuario
- Condiciones de entrada: El usuario se accede a la aplicación
- Condiciones de salida: La aplicación muestra la presentación de esta.
- Flujo de eventos:
 - ❖ El usuario accede a la aplicación
 - ❖ La aplicación muestra la información

Caso de Uso “Unirse a Sala”

- Actor: Usuario Registrado
- Condiciones de entrada: El usuario se encuentra registrado en la aplicación
- Condiciones de salida: El usuario se une a una sala ya existente
- Flujo de eventos:
 - ❖ El usuario decide unirse a una sala
 - ❖ El usuario consigue el identificador de la sala a la que quiere unirse
 - ❖ El usuario proporciona el identificador de la sala

- ❖ El sistema verifica la información y, en caso de que reconozca como válidos los datos, se redirecciona al usuario a la sala en cuestión.
- Excepciones:
 - ExceptionLobbyNotFound: El identificador proporcionado no corresponde a ninguna sala.
 1. La aplicación muestra el mensaje de error “La sala no existe”
 2. Vuelve a solicitar los datos al usuario
 - ExceptionLobbyNotValid: El identificador proporcionado no cuenta con el formato válido.
 1. La aplicación muestra el mensaje de error “El identificador no tiene un formato válido”
 2. Vuelve a solicitar los datos al usuario
 - ExceptionUserNotLoggedIn: El usuario no se encuentra identificado.
 1. La aplicación muestra el mensaje de error “Por favor identifíquese para poder unirse a una sala”
 2. Redirecciona al sitio de identificación

Caso de Uso “Crear Sala”

- Actor: Usuario Registrado
- Condiciones de entrada: El usuario crea una sala nueva
- Condiciones de salida: La aplicación redirecciona a la nueva sala.
- Flujo de eventos:
 - ❖ El usuario decide crear una sala nueva
 - ❖ La aplicación redirecciona al usuario a la nueva sala.

Caso de Uso “Añadir Canción”

- Actor: Usuario Registrado
- Condiciones de entrada: El usuario se encuentra registrado en la aplicación
- Condiciones de salida: El usuario añade una canción a la lista de una sala
- Flujo de eventos:
 - ❖ El usuario decide añadir una nueva canción
 - ❖ La aplicación solicita el enlace a la canción en cuestión
 - ❖ El usuario proporciona la información requerida
 - ❖ El sistema verifica la información y, en caso de que reconozca como válidos los datos, se añade la nueva canción a la lista.
- Excepciones:
 - ExceptionSongNotFound: La url proporcionada no lleva a ninguna canción.

1. La aplicación muestra mensaje de error “La canción no existe o no se puede encontrar en estos momentos.”
2. Vuelve a solicitar los datos del usuario

Caso de Uso “Borrar Canción”

- Actor: Usuario Registrado
- Condiciones de entrada: El usuario borra una canción en la lista de una sala
- Condiciones de salida: La aplicación borra la canción en cuestión.
- Flujo de eventos:
 - ❖ El usuario decide borrar una canción en lista
 - ❖ El usuario busca la canción a borrar
 - ❖ El usuario vota para borrar la canción
 - ❖ El resto de los usuarios de la sala votan para borrar
 - ❖ La aplicación borra la canción si la votación es positiva.

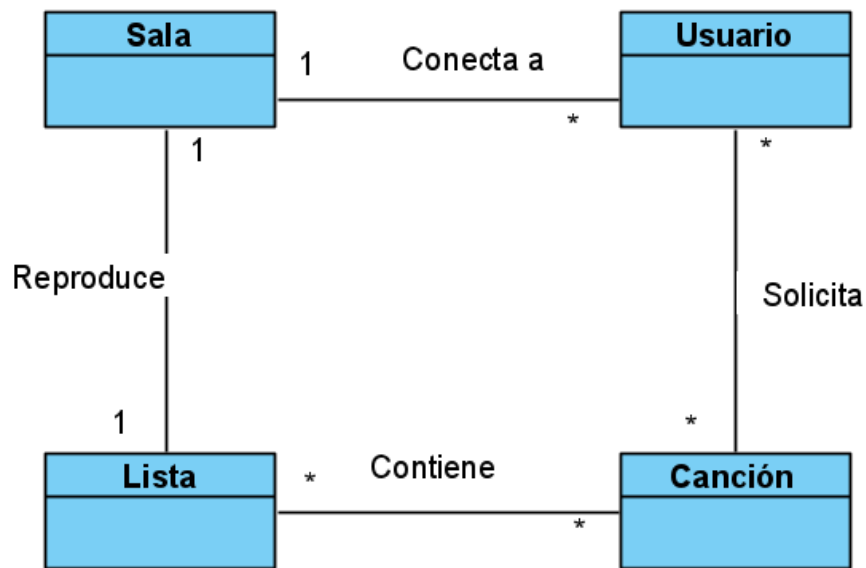
Caso de Uso “Cambiar Volumen”

- Actor: Usuario Registrado
- Condiciones de entrada: El usuario cambiar el volumen de su reproductor
- Condiciones de salida: La aplicación modula el volumen al gusto del usuario.
- Flujo de eventos:
 - ❖ El usuario decide cambiar el volumen de su reproductor
 - ❖ El usuario elije la cantidad usando el slider de volumen de la aplicación
 - ❖ La aplicación cambia el volumen del reproductor

Caso de Uso “Omitir Canción”

- Actor: Usuario Registrado
- Condiciones de entrada: El usuario omite la canción que él ha añadido
- Condiciones de salida: La canción es omitida por la aplicación.
- Flujo de eventos:
 - ❖ El usuario decide omitir una canción que él ha añadido
 - ❖ El usuario selecciona la canción en cuestión
 - ❖ La aplicación omite la canción.

Modelo de Dominio



En nuestro modelo del dominio hemos identificado las siguientes clases conceptuales: sala, usuario, lista y canción. Las relaciones existentes entre estas clases son las siguientes:

- ❖ Un usuario puede conectarse a una sola sala, mientras que en una sala puede haber varios usuarios
- ❖ En una sala solo puede reproducirse una lista simultáneamente
- ❖ Una lista puede contener varias canciones, y una canción puede estar en distintas listas simultáneamente.
- ❖ Un usuario puede solicitar una o varias canciones, y todos los usuarios pueden solicitar una canción.