

Documentación de clases Java de la Capa de Presentación



SCRABBLE

Grupo 41

Hongda Zhu (hongda.zhu@estudiantat.upc.edu)

Shenghao Ye (shenghao.ye@estudiantat.upc.edu)

Songhe Wang (songhe.wang@estudiantat.upc.edu)

Xuanyi Qiu (xuanyi.qiu@estudiantat.upc.edu)

1. Introducción	4
2. Capa de presentación	4
2.1 Controladores	4
2.1.1. PresentationController	4
2.1.2. ControladorConfiguracionView	4
2.1.3. ControladorDiccionarioView	5
2.1.4. ControladorJugadoresView	5
2.1.5. ControladorPartidaView	6
2.1.6. ControladorRankingView	6
2.1.7. MainViewController	7
2.2 Vistas	7
2.2.1. ConfigPartidaView	7
2.2.2 ConfiguracionAyudaView	8
2.2.3 ConfiguracionGeneralView	8
2.2.4 ConfiguracionPartidaView	9
2.2.5 CrearDiccionarioView	9
2.2.6 GestionDiccionariosView	10
2.2.7 GestionJugadoresView	10
2.2.8 GestionPartidaView	11
2.2.9 HistorialJugadorView	11
2.2.10 ModificarDiccionarioView	12
2.2.11 RankingView	12
2.2.12 VistaTablero	13
2.3 Componentes	14
2.3.1 CasillaDisplay	14
2.3.2 DiccionarioVisual	14
2.3.3 Ficha	14
2.3.4 JugadorRanking	15
2.3.5 MusicManager	15
2.4 Popups	15
2.4.1 customDialogo	15
2.4.2 AñadirLetraPopup	16
2.4.3 AñadirPalabraPopup	16
2.4.4 CambiarFichasPopup	16
2.4.5 CrearJugadorPopup	16
2.4.6 EliminarJugadorPopup	17
2.4.6 ImportarDiccionarioPopup	17
2.4.7 ModificarPalabraPopup	17
2.4.8 PausaPopup	18

1. Introducción

En este documento se encuentran las descripciones breves de todas las clases de la capa de presentación, al igual que otros datos significativos (como la cardinalidad, descripción de atributos y relaciones) que ayudan a la comprensión de la clase descrita.

Cabe destacar que la arquitectura empleada para la construcción de la capa de presentación es la de MVC (modelo-vista-controlador). Asimismo, véase que para cada vista le corresponde un controlador (exceptuando MainViewController [\(v. MainViewController\)](#) que carga una vista estática de un recurso fxml directamente ya que el menú principal no tiene ninguna complejidad dinámica y por lo tanto no es necesario crear una clase vista .java para éste). Además, se ha utilizado la librería 'javaFX' para la implementación de las interfaces gráficas, por la simpleza, estética y dinamismo que aporta en ellas.

En consecuencia a las decisiones tomadas, hemos pensado en seccionar la capa en 4 secciones principales para la optimización de la posterior organización: controladores, vistas, componentes y popups.

Los controladores son clases que permiten la navegación entre distintas vistas y manejan sus lógicas moviendo datos necesarios entre éstas y la capa de dominio. Las vistas por su parte son las interfaces en sí, pueden usar componentes reutilizables como: casillas [\(v. CasillaDisplay\)](#), fichas [\(v. Fichas\)](#), filas de ranking [\(v. JugadorRanking\)](#)... Y por último los popups son vistas auxiliares que se superponen a una pantalla inicial.

2. Capa de presentación

2.1 Controladores

2.1.1. PresentationController

- Breve descripción de la clase: Clase principal que actúa como intermediario entre la capa de presentación y el dominio. Centraliza todas las operaciones relacionadas con el juego, como gestión de partidas, jugadores, diccionarios y configuración. Implementa el patrón Singleton para garantizar una única instancia.
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:

- ctrlDomain: Instancia del controlador de dominio para comunicarse con la capa de dominio (static).

- presentationController: Instancia singleton del propio controlador de presentación (static).

- Descripción de las relaciones:

-

2.1.2. ControladorConfiguracionView

- Breve descripción de la clase: Controlador para la gestión de configuraciones generales de la aplicación. Maneja la navegación entre las vistas de configuración (general, partida y ayuda) y persiste las preferencias del usuario.

- Cardinalidad: una instancia única (singleton) compartida durante la ejecución de la aplicación

- Descripción de los atributos:

- stage: es la ventana principal
- presentationController: el controlador de presentación
- vistaAnterior: vista anterior desde la cual se accedió al módulo de configuración
- scene: es la escena principal
- vistaGeneral: vista de la configuración general
- vistaPartida: vista de la configuración partida
- vistaAyuda: vista de la página de ayuda
- vistaActual: vista actualmente mostrada

- Descripción de las relaciones:

- Relación de asociación con la clase “ConfiguracionGeneralView”: controlador que maneja la vista de la configuración general
- Relación de asociación con la clase “ConfiguracionAyudaView”: controlador que maneja la vista de la página de ayuda
- Relación de dependencia con la clase “PresentationController”: controlador de alto nivel con el que colabora este módulo
- Relación de asociación con la clase “ConfiguracionPartidaView” : controlador que maneja la vista de la configuración de partida

2.1.3. ControladorDiccionarioView

- Breve descripción de la clase: Controlador para la gestión de diccionarios del juego. Permite crear, modificar, eliminar e importar diccionarios, así como visualizar su contenido (alfabeto y palabras).

- Descripción de los atributos:

- presentationController: el controlador de presentación
- stage: es la ventana principal
- vistaAnterior: vista anterior desde la cual se accedió al módulo de configuración
- scene: es la escena principal
- layout: componente a mostrar
- diccionarios: lista de diccionarios visuales

- Descripción de las relaciones:

- Relación de dependencia con la clase “PresentationController”: controlador de alto nivel con el que colabora este módulo
- Relación de asociación con la clase “DiccionarioVisual”: diccionario visual utilizado para ser mostrado
- Relación de asociación con la clase “AñadirLetraPopup”: popup utilizado para añadir letra al abecedario del diccionario
- Relación de asociación con la clase “ModificarPalabraPopup”: popup utilizado para modificar una palabra del diccionario
- Relación de asociación con la clase “CrearDiccionariosView”: vista de crear diccionario
- Relación de asociación con la clase “ModificarDiccionariosView”: vista de modificar diccionario

2.1.4. ControladorJugadoresView

- Breve descripción de la clase: Controlador para la gestión de jugadores. Permite crear, eliminar y consultar estadísticas de jugadores, así como visualizar su historial de partidas.

- Cardinalidad: una instancia única compartida durante la ejecución de la aplicación

- Descripción de los atributos:

- layout: componente BorderPane que contiene dinámicamente las vistas
- presentationController: el controlador de presentación
- vistaAnterior: vista desde la que se accedió a gestión de jugadores
- jugadorSeleccionado: nombre del jugador actualmente seleccionado para ver su historial
- stage: ventana principal asociada al módulo de jugadores

- Descripción de las relaciones:

- Relación con la clase “GestionJugadoresView”: indica qué instancia de GestionJugadoresView gestiona
- Relación con la clase “HistorialJugadorView”: indica qué instancia de HistorialJugadoresView gestiona

-Relación con la clase “PresentationController”: indica qué controladora permite la conexión con la capa de dominio.

2.1.5. ControladorPartidaView

- Breve descripción de la clase: Controlador central para la gestión de partidas. Maneja la creación, carga y ejecución de partidas, así como la interacción con el tablero y los jugadores.
- Cardinalidad: Una instancia única para cada vista relacionada y la controladora de presentación.
- Descripción de los atributos:
 - vistaAnterior: la vista anterior desde la cual se accedió a gestión partidas
 - layout: contenedor principal de la vista
 - presentationController: la controladora de presentación
 - diccionario: diccionario seleccionado actualmente
 - size: tamaño del tablero configurado actualmente
 - jugadores: lista de jugadores seleccionados para jugar una partida
 - cargado: indica si la partida actual ha sido cargada o no
- Descripción de las relaciones:
 - Relación de asociación con la clase VistaTablero: indica qué instancia de VistaTablero se encarga.
 - Relación de asociación con la clase ConfigPartidaView: indica qué instancia de ConfigPartidaView se encarga.
 - Relación de dependencia con la clase GestionPartidaView: indica qué instancia de GestionPartidaView se encarga.
 - Relación de asociación con la clase PresentationController: indica qué controladora permite la conexión con la capa de dominio.

2.1.6. ControladorRankingView

- Breve descripción de la clase: Controlador para la gestión y visualización del ranking de jugadores. Muestra estadísticas de rendimiento (puntuaciones, victorias, partidas jugadas etc), permite eliminar jugadores del ranking al igual que filtrar y ordenar por diferentes parámetros.
- Cardinalidad: una instancia única compartida durante la ejecución de la aplicación

- Descripción de los atributos:

- stage: ventana principal asociada a la vista
- vistaAnterior: vista desde la que se accedió a la vista de ranking
- layout: contenedor principal de la vista
- instance: la instancia única del controlador de raking
- presentationController: el controlador de presentación

- Descripción de las relaciones:

- Relación de dependencia con la clase “PresentationController”: controlador de alto nivel con el que colabora este módulo
- Relación de asociación con la clase “EliminarJugadorPopup”: popup utilizado para eliminar jugador del ranking
- Relación de asociación con la clase “RankingView”: vista del ranking de jugadores

2.1.7. MainViewController

- Breve descripción de la clase: Controlador principal del menú de la aplicación. Gestiona la navegación a todas las secciones principales (jugadores, diccionarios, partidas, ranking, configuración).

- Cardinalidad: Una instancia única para cualquier clase relacionada

- Descripción de los atributos:

- btnGestionJugadores: botón que lleva a la pantalla de gestión de jugadores
- btnGestionDiccionarios: botón que lleva a la pantalla de gestión de diccionarios
- btnGestionPartidas: botón que lleva a la pantalla de gestión de partidas
- btnGestionRankings: botón que lleva a la pantalla de gestión de rankings
- btnConfiguracion: botón que lleva a la pantalla de gestión de configuración
- btnSalir: botón que permite salir de la aplicación
- presentationController: la controladora de presentación
- instance: la instancia única de MainViewController

- Descripción de las relaciones:

- Relación de asociación con la clase PresentationController para navegar o llevar datos del dominio en caso necesario

2.2 Vistas

2.2.1. ConfigPartidaView

- Breve descripción de la clase: Vista para configurar una nueva partida. Permite seleccionar diccionario, tamaño de tablero y jugadores participantes.
- Cardinalidad: Una o ninguna instancia de la clase ConfigPartidaView para ControladorPartidaView
- Descripción de los atributos:
 - ctrlPartida: controlador asociado para controlar la lógica de configuración
 - view: nodo raíz de la vista
 - txtTamanoTablero: campo de texto para definir tamaño del tablero
 - lblErrorTablero: texto en caso de tamaño incorrecto
 - btnIniciarPartida: botón para iniciar partida
 - btnVolver: botón para volver a la vista anterior
 - jugadoresDisponibles: caja que contiene todos los jugadores disponibles actuales
 - jugadoresSeleccionados: caja que contiene todos los jugadores seleccionados actuales
 - btnAgregarJugador: botón para agregar un jugador seleccionado
 - btnQuitarJugador: botón para eliminar un jugador de la caja de seleccionados
 - jugadorSeleccionado: botón del jugador seleccionado por un click actual
 - jugadoresReady: lista de los nombres de los jugadores disponibles a seleccionar
- Descripción de las relaciones:
 - Relación de asociación con “ControladorPartidaView”: indica la controladora que gestiona la lógica de la vista

2.2.2 ConfiguracionAyudaView

- Breve descripción de la clase: Vista de ayuda y contacto dentro de la configuración. Muestra información de soporte y permite navegar entre secciones.
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:
 - view: nodo raíz de la vista
 - btnGeneral: botón para navegar a la pantalla de configuración general

- btnPartida: botón para navegar a la pantalla de configuración de partida
 - btnAyuda: botón para navegar a la pantalla de ayuda
 - btnVolver: botón para regresar al menú principal
 - btnContacto: botón para mostrar información de contacto
- Descripción de las relaciones:

2.2.3 ConfiguracionGeneralView

- Breve descripción de la clase: Vista de configuración general. Permite seleccionar el idioma de la aplicación, el tema de la presentación, etc...
- Cardinalidad: Una o ninguna instancia de la clase ConfiguracionGeneral para ControladorConfiguracionView
- Descripción de los atributos:
 - view: nodo raíz de la vista
 - btnGeneral: botón para cambiar a la vista de configuración general
 - btnPartida: botón para cambiar a la vista de configuración de partida
 - btnAyuda: botón para cambiar a la vista de ayuda
 - btnVolver: botón para volver a la vista anterior
 - btnGuardar: botón para guardar cambios de configuración
 - btnRestablecer: botón para restablecer la configuración por defecto
 - toggleMusica: interruptor para activar o desactivar la música
 - toggleSonido: interruptor para activar o desactivar el sonido
 - lblEstadoMusica: etiqueta para indicar si la música está activada
 - lblEstadoSonido: etiqueta para indicar si el sonido está activado
 - toggleBgMusica: contenedor visual para el interruptor de música
 - toggleCircleMusica: círculo indicador del estado del interruptor de música
 - toggleBgSonido: contenedor visual para el interruptor de sonido
 - toggleCircleSonido: círculo indicador del estado del interruptor de sonido
 - sliderMusica: control deslizante para ajustar el volumen de la música
 - sliderSonido: control deslizante para ajustar el volumen del sonido
 - labelVolumenMusica: etiqueta que muestra el valor del volumen de música
 - labelVolumenSonido: etiqueta que muestra el valor del volumen de sonido
- Descripción de las relaciones:
 - Relación de asociación con “ControladorConfiguracionView”: indica la controladora que gestiona la lógica de la vista

2.2.4 ConfiguracionPartidaView

- Breve descripción de la clase: Clase ConfiguracionPartidaView para la simulación de la aplicación.
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:
 - view: Tipo Parent (no static)
 - btnGeneral: Tipo Button (no static)
 - btnPartida: Tipo Button (no static)
 - btnAyuda: Tipo Button (no static)
 - btnVolver: Tipo Button (no static)
 - btnGuardar: Tipo Button (no static)
 - btnRestablecer: Tipo Button (no static)
 - txtTamanoTablero: Tipo TextField (no static)
 - lblErrorTablero: Tipo Label (no static)
 - txtBots: Tipo TextField (no static)
 - lblErrorBots: Tipo Label (no static)
- Descripción de las relaciones:

2.2.5 CrearDiccionarioView

- Breve descripción de la clase: Clase que representa la vista para crear un nuevo diccionario. Esta vista permite al usuario ingresar el nombre del diccionario, el alfabeto y las palabras.
- Cardinalidad: Una o ninguna instancia de la clase CrearDiccionarioView para ControladorDiccionarioView
- Descripción de los atributos:
 - view: nodo raíz de la vista
 - controlador: controlador asociado para controlar la lógica de crear diccionario
 - btnCancelar: botón para cancelar la acción y volver a la vista anterior
 - btnConfirmar: botón para confirmar la acción y crear un diccionario
 - btnAñadirLetra: botón para añadir una letra al alfabeto
 - btnAñadirPalabra: botón para añadir una palabra al diccionario
 - headerPalabras: contenedor visual superior para la tabla de palabras
 - headerAlfabeto: contenedor visual superior para la tabla de alfabeto
 - nombreDiccionario: campo de texto donde se introduce el nombre del nuevo diccionario
 - alfabeto: tabla de letras que forman el alfabeto personalizado
 - tablaPalabras: tabla que contiene las palabras ingresadas por el usuario
 - palabrasOriginales: lista observable de palabras originales (inicial o temporal)

- alfabetoOriginal: lista observable del alfabeto original (estructura de listas de letras)
- Descripción de las relaciones:
 - Relación de asociación con la clase “ControladorDiccionarioView”: controlador que maneja la lógica a la creación de diccionario

2.2.6 GestionDiccionariosView

- Breve descripción de la clase: Vista encargada de la gestión de diccionarios. Permite al usuario crear, modificar, eliminar o importar diccionarios, así como visualizar una lista de diccionarios existentes en forma de tabla.
- Cardinalidad: Una o ninguna instancia de la clase GestionDiccionariosView para ControladorDiccionarioView
- Descripción de los atributos:
 - view: nodo raíz de la vista
 - controlador: controlador asociado para controlar la lógica de crear diccionario
 - btnCrear: botón para iniciar la creación de un nuevo diccionario
 - btnModificar: botón para modificar un diccionario seleccionado
 - btnEliminar: botón para eliminar un diccionario de la lista
 - btnImportar: botón para importar un diccionario desde un archivo
 - btnVolver: botón para volver a la vista anterior
 - listaDiccionarios: tabla que muestra los diccionarios disponibles
- Descripción de las relaciones:
 - Relación de asociación con la clase “ControladorDiccionarioView”: controlador utilizado para la lógica de interacción desde esta vista
 - Relación de asociación con la clase “DiccionarioVisual”: clase que representa visualmente cada diccionario en la tabla

2.2.7 GestionJugadoresView

- Breve descripción de la clase: Vista para administrar jugadores (crear, eliminar, ver historial).
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:
 - view: la vista principal
 - btnCrearJugador: botón para crear un jugador nuevo
 - btnVerHistorial: botón para ver historial de un jugador seleccionado

- btnEliminarJugador: botón para eliminar jugador
- btnVolver: botón para volver a la vista anterior
- txtBuscar: campo de texto de búsqueda
- btnBuscar: botón para realizar la búsqueda
- btnRestablecer: botón para restablecer la barra de búsqueda
- tablaJugadores: es la tabla que contiene todos los datos de los jugadores
- jugadorRow: clase privada que representa una fila de la tabla con los datos necesarios

- Descripción de las relaciones:

- Relación de asociación con “ControladorJugadoresView”: indica qué controladora gestiona la lógica de la vista
- Relación de dependencia con “CrearJugadorPopup”: indica que la vista usa momentáneamente esta clase para poder crear un jugador.

2.2.8 GestionPartidaView

- Breve descripción de la clase: Vista para gestionar partidas guardadas de Scrabble, permitiendo crear, cargar y eliminar partidas.
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:

- controlador: referencia al controlador de la vista de partida para manejar las operaciones.
- view: representa la vista FXML cargada para la gestión de partidas.
- btnJugarNuevaPartida: botón para iniciar una nueva partida.
- btnEliminarPartida: botón para eliminar una partida guardada.
- btnCargarPartida: botón para cargar una partida existente.
- btnVolver: botón para regresar al menú anterior.
- tablaPartidas: tabla que muestra las partidas guardadas.
- PartidaRow: clase privada que representa una fila de la partida con los datos necesarios

- Descripción de las relaciones:

- Relación de asociación con la clase “ControladorPartidaView”: indica la controladora que manejará la lógica de la vista

2.2.9 HistorialJugadorView

- Breve descripción de la clase: Vista para mostrar el historial detallado de un jugador, incluyendo sus estadísticas y puntuaciones.
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:

- controlador: Referencia al controlador de la vista de jugadores para manejar las operaciones.
- nombreJugador: Nombre del jugador cuyo historial se está mostrando.
- view: Representa la vista FXML cargada para el historial del jugador.
- lblNombreJugador: Etiqueta para mostrar el nombre del jugador.
- lblPuntuacionTotal: Etiqueta para mostrar la puntuación total del jugador.
- lblPuntuacionMaxima: Etiqueta para mostrar la puntuación máxima del jugador.
- lblPuntuacionMedia: Etiqueta para mostrar la puntuación media del jugador.
- lblPartidasJugadas: Etiqueta para mostrar el número de partidas jugadas.
- lblVictorias: Etiqueta para mostrar el número de victorias.
- lblRatioVictorias: Etiqueta para mostrar el ratio de victorias.
- listaHistorial: Lista para mostrar el historial de puntuaciones del jugador.
- btnVolver: Botón para regresar a la vista de gestión de jugadores.

- Descripción de las relaciones:

-Relación de asociación con la clase “ControladorJugadoresView”: indica qué controladora maneja la lógica de la vista.

2.2.10 ModificarDiccionarioView

- Breve descripción de la clase: Vista para modificar un diccionario existente. Permite al usuario consultar el alfabeto y actualizar las palabras de un diccionario seleccionado.
- Cardinalidad: Una o ninguna instancia de la clase ModificarDiccionarioView para ControladorDiccionarioView.

- Descripción de los atributos:

- view: nodo raíz de la vista
- controlador: controlador asociado para controlar la lógica de modificar diccionario
- campoNombre: campo de texto que muestra el nombre del diccionario (deshabilitado para edición)
- btnCancelar: botón para cancelar la acción y volver a la vista anterior
- btnGuardar: botón para confirmar la acción y modificar un diccionario
- btnAnadirPalabra: botón para añadir una nueva palabra al diccionario
- headerPalabras: contenedor visual superior para la tabla de palabras
- headerAlfabeto: contenedor visual superior para la tabla de alfabeto
- tablaAlfabeto: tabla de letras que forman el alfabeto personalizado
- tablaPalabras: tabla que contiene las palabras del diccionario
- palabrasOriginales: lista observable de palabras originales (inicial o temporal)
- alfabetoOriginal: lista observable del alfabeto original (estructura de listas de letras)

- Descripción de las relaciones:
 - Relación de asociación con la clase “ControladorDiccionarioView”: controlador utilizado para la lógica de interacción desde esta vista
 - Relación de asociación con la clase “DiccionarioVisual”: clase que representa visualmente cada diccionario

2.2.11 RankingView

- Breve descripción de la clase: Vista encargada de mostrar el ranking de jugadores en la aplicación. Permite filtrar y ordenar resultados, y eliminar jugador del ranking.
- Cardinalidad: Una o ninguna instancia de la clase RankingView para ControladorRankingView.

- Descripción de los atributos:
 - view: nodo raíz de la vista
 - controlador: controlador asociado para controlar la lógica de ranking
 - volverBtn: botón para regresar a la vista anterior
 - eliminarBtn: botón para eliminar un jugador del ranking
 - searchBtn: botón para buscar dentro del ranking
 - restablecerBtn: botón para restaurar el listado original
 - quantityField: campo de texto que permite definir el límite para el filtro
 - orderCombo: menú desplegable para ordenar el ranking según diferentes criterios
 - filterCombo: menú desplegable para aplicar filtro
 - table: tabla de tipo TableView<JugadorRanking> que muestra los jugadores
 - datosOriginales: lista observable con los datos completos del ranking
 - datosVisibles: lista observable que contiene los datos visibles según filtros/orden
 - errorLabel: etiqueta para mostrar mensajes de error al usuario
- Descripción de las relaciones:
 - Relación de asociación con la clase “ControladorRankingView”: controlador utilizado para la lógica de interacción desde esta vista
 - Relación de asociación con la clase “JugadorRanking”: clase que representa un jugador con sus estadísticas

2.2.12 VistaTablero

- Breve descripción de la clase: Clase que gestiona la interfaz gráfica del tablero de Scrabble, controlando la interacción del jugador, las reglas del juego y la visualización de la partida.
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:

- panelPrincipal: Panel principal de BorderPane que contiene los elementos de la interfaz.
- tablero: GridPane que representa el tablero de juego con las casillas.
- fichasJugador: HBox que muestra las fichas del jugador actual.
- jugadorActual: Label que muestra el nombre del jugador en turno.
- btnConfirmar: Botón para confirmar la jugada actual.
- btnCancelar: Botón para cancelar la jugada actual.
- btnPasarTurno: Botón para pasar el turno.
- puntuacionesContainer: VBox que muestra las puntuaciones de los jugadores.
- historialContainer: VBox que muestra el historial de jugadas.
- lblFichasRestantes: Label que indica la cantidad de fichas restantes en la bolsa.

- scrollPane: ScrollPane que contiene el tablero para desplazamiento.
- filasNumeracion: VBox para mostrar la numeración de filas.
- columnasNumeracion: HBox para mostrar la numeración de columnas.
- btnCambiarFichas: Botón para cambiar fichas.
- lblDiccionarioSeleccionado: Label que muestra el diccionario de la partida.
- controlador: referencia al controlador de la vista de partida para manejar la lógica del juego.
- fichaSeleccionada: la ficha seleccionada actual
- view: la vista fxml cargada
- N: tamaño del tablero
- saved: indica si la partida ha sido guardada
- firstMove: indica si es la primera jugada de la partida
- jugadorActualNombre: nombre del jugador actual
- direccionActual: la dirección en la que se están colocando las fichas actuales
- FichaColocada: clase privada que representa una ficha con una tupla de datos justos y necesarios.
- historialJugadas: lista para mantener un registro de las jugadas realizadas.
- fichasColocadasEnTurnoActual: Lista de fichas colocadas en el turno actual.
- posiciones: Mapa que registra las letras colocadas en cada posición del tablero.
- CASILLA_SIZE: Tamaño fijo para cada casilla del tablero.
- jugadoresPuntuaciones: Mapa que almacena las puntuaciones de cada jugador.
- rackActual: Mapa que representa las fichas del jugador actual.
- jugadores: Lista de nombres de jugadores en la partida.
- jugadorActualIndex: Índice del jugador actual en la lista de jugadores.

• Descripción de las relaciones:

- Relación de asociación con la clase “ControladorPartidaView”: indica qué controladora gestiona la lógica de la vista

- Relación de asociación con la clase “Ficha”: indica las fichas que pueden haber en una partida
- Relación de dependencia con la clase “CasillaDisplay”: indica el uso momentáneo de la casilla para crear y gestionar.
- Relación de dependencia con la clase “PausaPopup”: indica el uso momentáneo del menú de pausa al pausar la partida

2.3 Componentes

2.3.1 CasillaDisplay

- Breve descripción de la clase: Componente personalizado que representa una ficha con su letra, puntos y estado de selección.

- Descripción de los atributos:

- tipo: tipo de casilla especial (TipoCasilla enum) que determina el color y texto de la casilla
- fila: posición vertical de la casilla en el tablero (coordenada Y)
- columna: posición horizontal de la casilla en el tablero (coordenada X)
- casillaLabel: etiqueta que muestra el texto de la casilla
- fichaContainer: contenedor visual que almacena todos los elementos de la ficha colocada
- fichaLabel: etiqueta que muestra la letra de la ficha cuando está colocada
- puntosLabel: etiqueta que muestra los puntos de la ficha en la esquina inferior derecha
- tieneFicha: booleano que indica si hay una ficha colocada en esta casilla
- fichaConfirmada: booleano que indica si la ficha colocada ya fue confirmada/jugada
- bordeResaltado: rectángulo naranja que se muestra cuando la casilla puede recibir una ficha
- tamano: tamaño en píxeles de la casilla (ancho y alto), por defecto 40px

- Descripción de las relaciones:

2.3.2 DiccionarioVisual

- Breve descripción de la clase: Clase que representa un diccionario en forma visual para su uso en la capa de presentación. Contiene el nombre del diccionario, el alfabeto que lo compone y la lista de palabras asociadas.
- Cardinalidad: Una instancia por cada diccionario visualizado en las vistas de gestión, creación o modificación.

- Descripción de los atributos:
 - nombre: nombre del diccionario
 - alfabeto: lista de letras permitidas en el diccionario
 - palabras: lista de palabras válidas para ese diccionario
- Descripción de las relaciones:

2.3.3 Ficha

- Breve descripción de la clase: Clase que representa una ficha visual, muestra la letra, puntos y maneja estados de selección con animaciones de elevación.
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:
 - letra: Carácter que representa la letra de la ficha (String)
 - puntos: Valor numérico de puntos que otorga la ficha al jugarla
 - seleccionada: Booleano que indica si la ficha está actualmente seleccionada por el jugador
 - fichaFondo: Rectángulo que forma el fondo visual de la ficha con bordes redondeados
 - letraLabel: Etiqueta central que muestra la letra de la ficha en grande
 - puntosLabel: Etiqueta pequeña en esquina inferior derecha que muestra los puntos
 - tamanoFicha: Tamaño en píxeles de la ficha (ancho y alto), por defecto 40px
 - ELEVACION_ALTURA: Constante que define cuántos píxeles se eleva la ficha al seleccionarla (-10px)
- Descripción de las relaciones:

2.3.4 JugadorRanking

- Breve descripción de la clase: Clase que representa un jugador dentro del ranking de la aplicación. Contiene sus estadísticas relevantes como nombre, puntuación total, puntuación máxima, promedio de puntos, cantidad de partidas jugadas y número de victorias.
- Cardinalidad: Una instancia por cada jugador visualizado en el ranking
- Descripción de los atributos:
 - nombre: nombre del jugador
 - puntosTotales: puntos acumulados en todas las partidas
 - puntosMaximos: puntuación más alta obtenida en una sola partida

- puntosMedias: promedio de puntos por partida
- partidas: número total de partidas jugadas
- victorias: número de partidas ganadas

- Descripción de las relaciones:

2.3.5 MusicManager

- Breve descripción de la clase: Clase utilitaria que gestiona la reproducción de música en la aplicación. Encapsula la inicialización, reproducción, pausa y ajuste de volumen mediante MediaPlayer de JavaFX. Está diseñada como una clase estática y centralizada, sin instancias.
- Cardinalidad: Clase singleton mediante static: no se instancian objetos, sino que se usa a nivel global durante toda la ejecución de la aplicación.
- Descripción de los atributos:
 - mediaPlayer: reproductor de medios para reproducir la música de fondo
 - initialized: booleano que indica si el reproductor está inicializado o no

- Descripción de las relaciones:

2.4 Popups

2.4.1 customDialogo

- Breve descripción de la clase: Clase utilitaria que proporciona métodos estáticos para mostrar diálogos de alerta personalizados según el tipo
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:
 - AlertType (enum): Define cuatro tipos de alerta (INFO, WARNING, ERROR, SUCCESS) con colores, títulos e iconos asociados
 - ICON_MAP: Mapa estático que asocia cada tipo de alerta con su emoji correspondiente
- Descripción de las relaciones:

2.4.2 AñadirLetraPopup

- Breve descripción de la clase: Clase AñadirLetraPopup representa un cuadro de diálogo modal que permite al usuario añadir una nueva letra al alfabeto de un diccionario.

El popup incluye validación de entrada para evitar duplicados o caracteres inválidos.

- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:
 - alfabeto: lista observable de listas que representa el alfabeto al que se añadirá una nueva letra.
- Descripción de las relaciones:

2.4.3 AñadirPalabraPopup

- Breve descripción de la clase: Clase AñadirPalabraPopup representa un cuadro de diálogo modal que permite al usuario añadir una nueva palabra a un diccionario. Incluye validaciones para evitar palabras duplicadas o con caracteres no válidos, y asegura que la palabra esté formada por letras presentes en el alfabeto del diccionario.
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:
 - palabras: lista observable que contiene las palabras actuales del diccionario
 - alfabeto: conjunto con las letras válidas del diccionario
 - controlador: controlador asociado para controlar la lógica del popup
- Descripción de las relaciones:

2.4.4 CambiarFichasPopup

- Breve descripción de la clase: Popup modal para permitir al jugador seleccionar fichas de su rack para cambiarlas.
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:
 - stage: Ventana modal del popup que bloquea la interacción con otras ventanas
 - fichasSeleccionadas: Lista de objetos Ficha que el usuario ha seleccionado para cambiar
 - letrasSeleccionadas: Lista de strings con las letras correspondientes a las fichas seleccionadas
 - lblSeleccionadas: Etiqueta que muestra el contador de fichas seleccionadas ("X seleccionadas")
 - casillaTamaño: Tamaño en píxeles que tendrán las fichas mostradas en el popup
 - vistaTablero: Referencia al tablero para obtener los puntos de cada letra

- Descripción de las relaciones:

2.4.5 CrearJugadorPopup

- Breve descripción de la clase: Popup modal para crear un nuevo jugador en el sistema. Solicita el nombre del jugador y valida que sea único antes de crearlo.

- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.

- Descripción de los atributos:

- controlador: referencia al controlador de la vista de jugadores para realizar las operaciones

- Descripción de las relaciones:

-

2.4.6 EliminarJugadorPopup

- Breve descripción de la clase: Clase EliminarJugadorPopup representa un cuadro de diálogo personalizado que solicita confirmación para eliminar un jugador del ranking de la aplicación.

- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.

- Descripción de los atributos:

- popupStage: ventana modal que representa el diálogo

- Descripción de las relaciones:

2.4.6 ImportarDiccionarioPopup

- Breve descripción de la clase: Clase ImportarDiccionarioPopup representa una ventana modal que permite al usuario especificar la ruta de un directorio desde el cual se desea importar un diccionario.

- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.

- Descripción de los atributos:

- Descripción de las relaciones:

2.4.7 ModificarPalabraPopup

- Breve descripción de la clase: Clase ModificarPalabraPopup representa un cuadro de diálogo modal que permite modificar una palabra existente dentro de un diccionario. Incluye validaciones sobre la entrada del usuario para evitar duplicados o letras inválidas.
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:
 - palabras: lista observable que contiene las palabras actuales del diccionario
 - alfabeto: conjunto con las letras válidas del diccionario
 - palabraOriginal: palabra que se desea modificar
 - controlador: controlador asociado para controlar la lógica del popup
- Descripción de las relaciones:
 - Relación de agregación con la clase “ControladorDiccionarioView”: controlador utilizado para mostrar alertas

2.4.8 PausaPopup

- Breve descripción de la clase: Vista que aparece cuando se para una partida
- Cardinalidad: Una instancia por cada uso según el contexto de la aplicación.
- Descripción de los atributos:
 - ButtonStyle (enum): Define cinco estilos visuales para botones (SUCCESS, INFO, WARNING, DANGER, SECONDARY) con sus clases CSS correspondientes
 - PopupButton (clase interna): Encapsula la configuración de un botón del popup con texto, estilo visual y función callback
- Descripción de las relaciones: