Listado de Jugadores.

El tema elegido es el guardado de datos de jugadores de distintos deportes.

El programa permitirá :

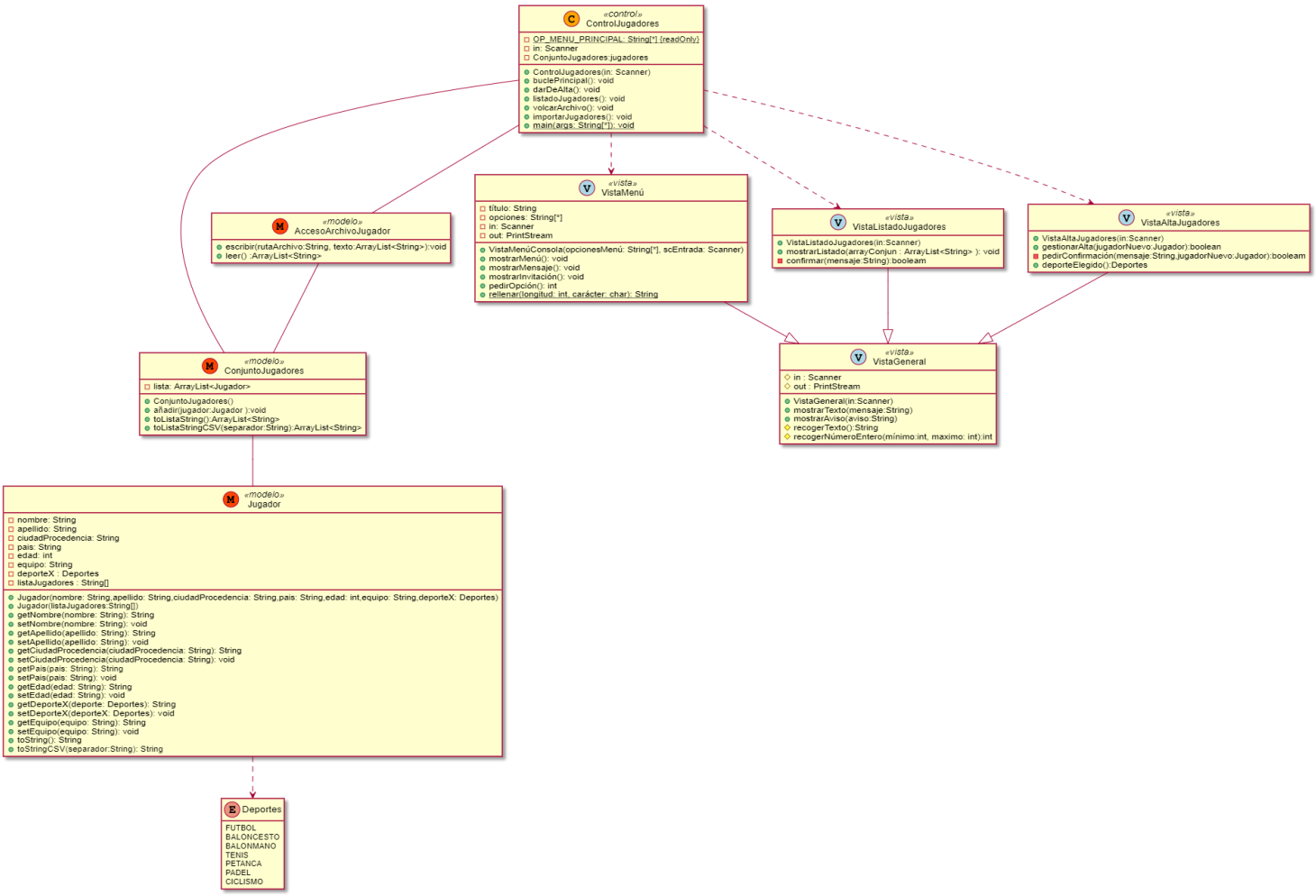
-Dar de alta a los distintos jugadores con sus respectivos datos.

-Ver el listado de los Jugadores guardados.

-Exportar todos los jugadores a un archivo .txt.

-Importar desde el archivo a los jugadores exportados anteriormente o añadidos a mano al archivo.

Diagrama UML



CODIGO PLANTUML

```
@startuml

hide empty members


class ControlJugadores <<(C,orange) control>>{

- {static} OP_MENU_PRINCIPAL: String[*] {readOnly}

- in: Scanner

- ConjuntoJugadores:jugadores

+ ControlJugadores(in: Scanner)

+ buclePrincipal(): void

+ darDeAlta(): void

+ listadoJugadores(): void

+ volcarArchivo(): void

+ importarJugadores(): void

+ {static} main(args: String[*]): void

}



class ConjuntoJugadores <<(M,orangered) modelo>>{

- lista: ArrayList<Jugador>

+ConjuntoJugadores()

+añadir(jugador:Jugador ):void

+toListaString():ArrayList<String>

+toListaStringCSV(separador:String):ArrayList<String>

}


class Jugador <<(M,orangered) modelo>>{
```

- nombre: String

- apellido: String

- ciudadProcedencia: String

- pais: String

- edad: int

- equipo: String

- deporteX : Deportes

- listaJugadores : String[]

+ Jugador(nombre: String,apellido: String,ciudadProcedencia: String,pais: String,edad: int,equipo: String,deporteX: Deportes)

+ Jugador(listaJugadores:String[])

+ getNombre(nombre: String): String

+ setNombre(nombre: String): void

+ getApellido(apellido: String): String

+ setApellido(apellido: String): void

+ getCiudadProcedencia(ciudadProcedencia: String): String

+ setCiudadProcedencia(ciudadProcedencia: String): void

+ getPais(pais: String): String

+ setPais(pais: String): void

+ getEdad(edad: String): String

+ setEdad(edad: String): void

+ getDeporteX(deporte: Deportes): String

+ setDeporteX(deporteX: Deportes): void

+ getEquipo(equipo: String): String

+ setEquipo(equipo: String): void

+ toString(): String

+ toStringCSV(separador:String): String

}

enum Deportes{

FUTBOL

BALONCESTO

BALONMANO

TENIS

PETANCA

PADEL

CICLISMO

}

class VistaGeneral <<(V,lightblue) vista>>{

in : Scanner

out : PrintStream

+VistaGeneral(in:Scanner)

+mostrarTexto(mensaje:String)

+mostrarAviso(aviso:String)

#recogerTexto():String

#recogerNúmeroEntero(mínimo:int, maximo: int):int

}

class VistaMenú <<(V,lightblue) vista>>{

- título: String

- opciones: String[*]

- in: Scanner

- out: PrintStream

+ VistaMenúConsola(opcionesMenú: String[*], scEntrada: Scanner)

+ mostrarMenú(): void

+ mostrarMensaje(): void

+ mostrarInvitación(): void

+ pedirOpción(): int

+ {static} rellenar(longitud: int, carácter: char): String

}

class VistaListadoJugadores <<(V,lightblue) vista>>{

+VistaListadoJugadores(in:Scanner)

+ mostrarListado(arrayConjun : ArrayList<String>): void

-confirmar(mensaje:String):booleam

}

class VistaAltaJugadores <<(V,lightblue) vista>>{

+VistaAltaJugadores(in:Scanner)

+ gestionarAlta(jugadorNuevo:Jugador):boolean

-pedirConfirmación(mensaje:String,jugadorNuevo:Jugador):booleam

+deporteElegido():Deportes

}

class AccesoArchivoJugador <<(M,orangered) modelo>>{

+ escribir(rutaArchivo:String, texto:ArrayList<String>):void

+ leer() :ArrayList<String>

}

ControlJugadores ..> VistaMenú

ControlJugadores ..> VistaListadoJugadores

ControlJugadores ..> VistaAltaJugadores

VistaMenú --|> VistaGeneral

VistaListadoJugadores--|> VistaGeneral

VistaAltaJugadores --|> VistaGeneral

ConjuntoJugadores -- Jugador

ControlJugadores -- ConjuntoJugadores

Jugador ..> Deportes

AccesoArchivoJugador -- ConjuntoJugadores

ControlJugadores -- AccesoArchivoJugador

@enduml