**LABORATORIO APO II**

**UNIDAD 4 LABORATORIO 2 - ANÁLISIS DE REQUERIMIENTOS E IMPLEMENTACIÓN**

*LIZARAZO RANGEL AGUSTÍN(A00311880) -CORRALES MEJÍA GUSTAVO ANDRÉS (A00012136)*

1. **Análisis de requerimientos**

|  |  |
| --- | --- |
| **Nombre:** | R1. Seleccionar tipo de entrenamiento. |
| **Resumen:** | El usuario podrá seleccionar un tipo de entrenamiento a efectuar dentro del gimnasio. En el entrenamiento de tipo 1 el usuario verá a su pokemon correr hasta una línea de meta y en el entrenamiento tipo 2 el usuario practicará atrapar a su pokemon dentro de la pokebola mientras corre hacía y antes de que toque la línea de meta. |
| **Entradas:** | Clic sobre la opción de entrenamiento seleccionada |
| **Resultado:** | El entrenamiento del tipo seleccionado se ha instanciado para iniciar. |

|  |  |
| --- | --- |
| **Nombre:** | R2. Seleccionar el pokémon elegido. |
| **Resumen:** | El usuario podrá seleccionar un pokémon dentro de una lista de éstos para así hacer su entrenamiento con él. |
| **Entradas:** | Clic izquierdo sobre el pokémon elegido, imágenes pokemones |
| **Resultado:** | El pokémon ha sido creado dentro del arreglo de pokemons |

|  |  |
| --- | --- |
| **Nombre:** | R3. Generar movimiento de pokemon |
| **Resumen:** | El usuario seleccionará la pokebola dentro de la cual se encuentra el pokémon, al momento de seleccionarla el pokémon saldrá de la pokebola y empezará a correr hasta la bandera de llegada, donde se detendrá. |
| **Entradas:** | Clic izquierdo sobre la pokebola, imagen pokemon, imagen pokebola |
| **Resultado:** | La imagen de la pokebola se ha intercambiado por la de un pokemon y el pokémon ha empezado a desplazarse de izquierda a derecha hasta la posición de la bandera. |

|  |  |
| --- | --- |
| **Nombre:** | R4. Detener movimiento de pokemon |
| **Resumen:** | El pokémon se desplazará hasta la posición de la bandera donde se detendrá. |
| **Entradas:** | posX de la bandera y posX del pokemon |
| **Resultado:** | El pokémon termina su recorrido. El hilo que se encarga del movimiento del pokémon se para. |

|  |  |
| --- | --- |
| **Nombre:** | R5. Mostrar distancia recorrida por el pokémon |
| **Resumen:** | El pokémon al llegar a la línea de llegada dejará de moverse y se generará una alerta que indica la cantidad desplazada por el pokémon desde su inicio. |
| **Entradas:** | Distancia recorrida por el pokémon |
| **Resultado:** | Una alerta que diga “Su pokemon se ha desplazado \*\*\*\*\* pixeles en su entrenamiento” |

|  |  |
| --- | --- |
| **Nombre:** | R6. Atrapar pokémon en un recorrido |
| **Resumen:** | Si el usuario selecciona el tipo de entrenamiento 2 tendrá la facultad de atrapar a su pokemon mientras este se dirige hacía la línea de meta, si el usuario hace click sobre su pokemon antes de que logre llegar a su meta el pokémon entra a su pokebola |
| **Entradas:** | Click izquierdo sobre el pokémon, imagen de pokebola |
| **Resultado:** | El pokemon ha terminado su recorrido (su hilo se detiene) y la imagen del pokemon es intercambiada por una pokebola. pokemonCapturado==true; |

|  |  |
| --- | --- |
| **Nombre:** | R7. Guardar nombre del pokémon capturado en un archivo txt |
| **Resumen:** | Si el usuario está en el tipo de entrenamiento 2 y el pokemon ha sido capturado se mostrará una alerta que le pedirá al usuario mediante un textField el nombre del pokemon capturado para así guardarlo en un archivo txt. |
| **Entradas:** | pokemonCapturado==true, click en el botón de guardar nombre |
| **Resultado:** | Se muestra una alerta que indica que el usuario ha capturado un pokémon y le pide mediante un textField el nombre del nuevo pokémon capturado que se guardará en un archivo de texto. |

|  |  |
| --- | --- |
| **Nombre:** | R8. Modificar aleatoriamente la velocidad del pokemon en los dos entrenamientos |
| **Resumen:** | Si el usuario está en el tipo de entrenamiento 2 o 1, la velocidad base de cada pokemon se modificará aleatoriamente oscilando en un valor randomico |
| **Entradas:** | - |
| **Resultado:** | El pokemon se desplaza con una velocidad cambiante en los dos tipos de entrenamiento |

|  |  |
| --- | --- |
| **Nombre:** | R9. Generar puntaje de pokemon en los dos tipos de entrenamiento |
| **Resumen:** | Si el usuario está en el tipo de entrenamiento 2 el pokemon tendrá un puntaje que descenderá a medida que el usuario se demore más en atrapar al pokemon y si el usuario está en el tipo de entrenamiento 1 el pokemon tendrá un puntaje ascendente que subirá a medida que el pokemon recorre más kilometros. |
| **Entradas:** | Tipo de entrenamiento |
| **Resultado:** | El pokemon suma o resta puntaje dependiendo del tipo de entrenamiento |

|  |  |
| --- | --- |
| **Nombre:** | R10. Visualizar puntaje en cada tipo de entrenamiento |
| **Resumen:** | Si el usuario está en el tipo de entrenamiento 2 o 1, debe poder ver su puntaje incrementando o decrementando para así saber cuánto lleva acumulado antes de guardar |
| **Entradas:** | puntaje jugador |
| **Resultado:** | El puntaje del jugador se visualiza en la pantalla sin importar el tipo de entrenamiento en el que esté el usuario |

|  |  |
| --- | --- |
| **Nombre:** | R11. Registrar nickname del jugador |
| **Resumen:** | Un jugador nuevo debe poder registrar algún nickname para iniciar el juego |
| **Entradas:** | nombre jugador |
| **Resultado:** | El jugador ha sido registrado en la base de datos exitosamente |

|  |  |
| --- | --- |
| **Nombre:** | R12. Guardar estado del juego |
| **Resumen:** | Un jugador ya registrado y en un entrenamiento debe poder guardar el estado actual de su juego en cualquier momento en el disco duro del computador |
| **Entradas:** | click en el botón guardar |
| **Resultado:** | El estado actual del juego del jugador “nick name” ha sido guardado con un puntaje de “puntaje obtenido hasta haber guardado” en el disco duro del computador |

|  |  |
| --- | --- |
| **Nombre:** | R13. Cargar estado del juego guardado |
| **Resumen:** | Un jugador debe poder cargar un estado de juego dando click en el botón de cargar e ingresando el nick del estado del juego del usuario que desea cargar |
| **Entradas:** | nick del jugador del estado de juego que se quiere cargar |
| **Resultado:** | El estado de guardado del jugador introducido es mostrado en pantalla |

|  |  |
| --- | --- |
| **Nombre:** | R14. Mostrar lista de jugadores en el juego por sus puntajes |
| **Resumen:** | Un jugador debe poder visualizar la lista de jugadores (al menos 15) que han jugado el juego en cualquiera de los dos tipos de entrenamiento ordenándolos de acuerdo a su puntaje obtenido de menor a mayor |
| **Entradas:** | click en el botón guardar |
| **Resultado:** | Una nueva ventana muestra todos los jugadores que han jugado el juego de manera ordenada con sus puntajes de menor a mayor |

|  |  |
| --- | --- |
| **Nombre:** | R15. Mostrar lista de jugadores en el juego de acuerdo a su nombre |
| **Resumen:** | Un jugador debe poder visualizar la lista de jugadores (al menos 15) que han jugado el juego en cualquiera de los dos tipos de entrenamiento ordenándolos de acuerdo a su nombre de manera alfabética |
| **Entradas:** | click en el botón guardar, click en el botón ordenar nombres en la ventana de jugadores mostrada después de guardar. |
| **Resultado:** | La ventana de jugadores mostrada después de guardar es organizada de acuerdo al nombre de todos los jugadores. |

|  |  |
| --- | --- |
| **Nombre:** | R16. Buscar a un jugador mediante su nombre utilizando la búsqueda binaria |
| **Resumen:** | Un jugador debe poder buscar entre los jugadores que ya han jugado a uno de ellos, para conocer su nombre y su puntaje utilizando la búsqueda binaria |
| **Entradas:** | Nick del jugador a buscar |
| **Resultado:** | Nombre del jugador buscado y su puntaje |

|  |  |
| --- | --- |
| **Nombre:** | R17. Mostrar un mensaje de excepción si se busca de manera binaria a un jugador inexistente |
| **Resumen:** | Un jugador debe poder buscar entre los jugadores que ya han jugado a uno de ellos, para conocer su nombre y su puntaje utilizando la búsqueda binaria, pero si introduce un nombre de un jugador inexistente entonces el sistema mostrará un mensaje de error indicando que no hay ningún jugador con dicho nombre. |
| **Entradas:** | Nick del jugador a buscar |
| **Resultado:** | Mensaje de excepción no hay ningún jugador con dicho nombre. |

|  |  |
| --- | --- |
| **Nombre:** | R18. Buscar algún pokemon por el nombre que se le pase por parámetro. |
| **Resumen:** | Un jugador debe poder buscar el nombre de un pokemon si existe dentro de los pokemones existentes en el programa se mostrará. |
| **Entradas:** | Nick del pokemon a buscar |
| **Resultado:** | Nombre del pokemon |

|  |  |
| --- | --- |
| **Nombre:** | R19. Mostrar un mensaje de excepción si buscas algún pokémon por el nombre que no exista en el programa |
| **Resumen:** | Un jugador debe poder buscar el nombre de un pokemon si no existe dentro de los pokemones existentes, el programa mostrará un mensaje de excepción que indicará que el nombre de pokemon ingresado no existe |
| **Entradas:** | Nick del pokemon a buscar |
| **Resultado:** | Mensaje de excepción de que el nick de pokemon ingresado no existe |

**2. Trazabilidad**

Los requerimientos están listados en orden de desarrollo de los mismos. Aunque hay métodos que soportan varios requerimientos, se dejaron relacionados con el primer requerimiento desarrollado que lo requirió.

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R1. Seleccionar tipo de entrenamiento. | seleccionPokemon():void | Main |
| seleccionPokemon2():void |
|  |
| seleccionarPokemon(int):Pokemon | Campo |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R2. Seleccionar el pokémon elegido. | mostrar():void | SeleccionController |
| darIndices(Button):int |
|  | mostrar():void | SeleccionController2 |
|  | darIndices(Button):int |
|  | pokemonSeleccionado(int):void | Main |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R3. Generar movimiento de pokemon. | cambiarDesplazamientoPokemon(ImageView):void | CampoController |
|  | run():void | HiloCampoController |
|  | run():void | HiloCampoController2 |
|  | cambiarDesplazamientoPokemon2(ImageView):void | CampoController2 |
|  | cambioDesplazamiento() | Main |
|  | desplazamiento(long):void  desplazamiento2(long):void | Campo |
|  | run():void | HiloCampo |
|  | run():void | HiloCampo2 |
|  | cambiarPosicion(double):void | Pokemon |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R4. Detener movimiento de pokemon y R5 Mostrar distancia recorrida por el pokemon. | liberarPokemon(MouseEvent):void | CampoController |
|  | LlegoAlaMeta():boolean  pokemonLiberado(CampoController,ImageView):void  aviso():void | Main |
|  | desplazamiento(long): void | Campo |
|  | cambiarPosicion(double):void | Pokemon |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R6. Atrapar a pokémon en un recorrido y R7. Guardar nombre del pokémon capturado en un archivo txt | atraparPokemon(MouseEvent):void | CampoController2 |
|  | pokemonAtrapado():boolean  pokemonAtrapador(CampoController2,ImageView):void | Main |
|  | cambiarPokemonAtrapado(boolean):void  desplazamiento2(long):void | Campo |
|  | atrapado(boolean):void  cambiarPosicion(double):void | Pokemon |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R8. Modificar aleatoriamente la velocidad del pokemon en los dos entrenamientos | desplazamiento(long):void  desplazamiento2(long):void | Campo |
|  | velocidadAleatoria():void | Pokemon |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R9. Generar puntaje de pokemon en los dos tipos de entrenamiento | desplazamiento(long):void  desplazamiento2(long):void | Campo |
|  | bajarPuntaje():void  aumentarPuntaje():void | Pokemon |
|  | darPuntaje():int  cambiarPuntaje(int):void | Jugador |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R10. Visualizar puntaje en cada tipo de entrenamiento | aumentarPuntaje():void  cambiarDesplazamientoPokemon():void | campoController |
|  | aumentarPuntaje():void  cambiarDesplazamientoPokemon2():void | campoController2 |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R7 Guardar nombre del pokémon capturado en un archivo txt y R11. Registrar nickname del jugador y R12. Guardar estado del juego | guardarNick(ActionEvent):void | identificacionController |
|  | nueJugador(String):void | Main |
|  | nuevoJugador(String):void  guardar():void | Campo |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R13. Cargar estado del juego guardado | cargar(ActionEvent):void | identificacionController |
|  | cargar():void | Main |
|  | cargar():void | Campo |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R14. Mostrar lista de jugadores en el juego por sus puntajes | iniciar():void | puntajeController |
|  | getValoresDeJugadores():String[][] | Main |
|  | getValores():String[][]  guardar():void | Campo |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R15. Mostrar lista de jugadores en el juego de acuerdo a su nombre | iniciar():void  alfabeto():int[][]  valorOrden(String):double  buscarValorLetra(char,int):double  cambiarPosicionBoton(int):void  darPosicionBoton():int | puntajeController |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| R15. Mostrar lista de jugadores en el juego de acuerdo a su nombre | iniciar():void  alfabeto():int[][]  valorOrden(String):double  buscarValorLetra(char,int):double  cambiarPosicionBoton(int):void  darPosicionBoton():int | puntajeController |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| **R16.** Buscar a un jugador mediante su nombre utilizando la búsqueda binaria | cargarBinarioNombre(): void | Campo |
| busquedaBinariaPokemon(ActionEvent): void | IdentificacionController |
| cargarBusquedaBinaria():void | main |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| **R17.** Mostrar un mensaje de excepción si se busca de manera binaria a un jugador inexistente | **cargarBinarioNombre(): void** | **Campo** |
|  | **cargarBusquedaBinaria():void** | **Main** |
|  | **busquedaBinaria(ActionEvent):void** | **IdentificacionController** |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| **R18**. Buscar algún pokemon por el nombre que se le pase por parámetro. | **cargarBinarioPokemon():int** | **Campo** |
|  | **cargarBusquedaBinariaPokemon()():int** | **Main** |
|  | **busquedaBinariaPokemon(ActionEvent): void** | **IdentificacionController** |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Método** | **Clase** |
| **R19**. Mostrar un mensaje de excepción si buscas algún pokémon por el nombre que no exista en el programa | **cargarBinarioPokemon():int** | **Campo** |
|  | **cargarBusquedaBinariaPokemon()():int** | **Main** |
|  | **busquedaBinariaPokemon(ActionEvent): void** | **IdentificacionController** |

**3. Diseño de pruebas métodos de ordenamiento**

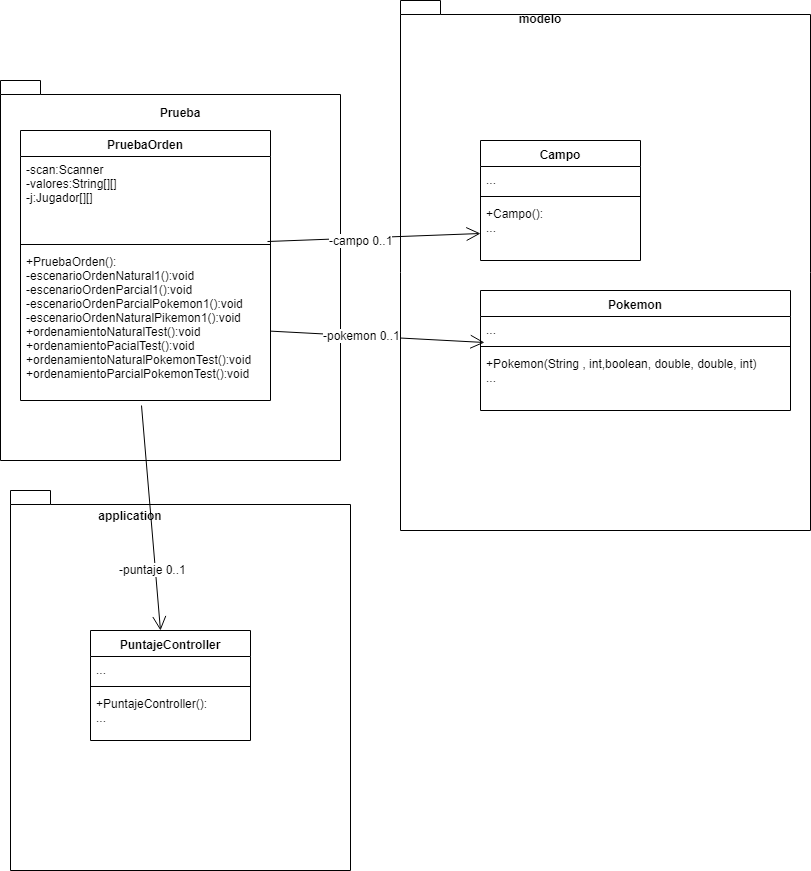
**P1.** Probar que el método guardar() de la clase campo ordena a los jugadores en una lista de acuerdo a su puntaje de manera natural del menor al mayor.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
| Campo | guardar(); | nuevoJugador2(“abril”,0);  nuevoJugador2(“Carlos”,50); | 0 | **Los jugadores han sido ordenados de acuerdo a su puntaje de menor a mayor.** |

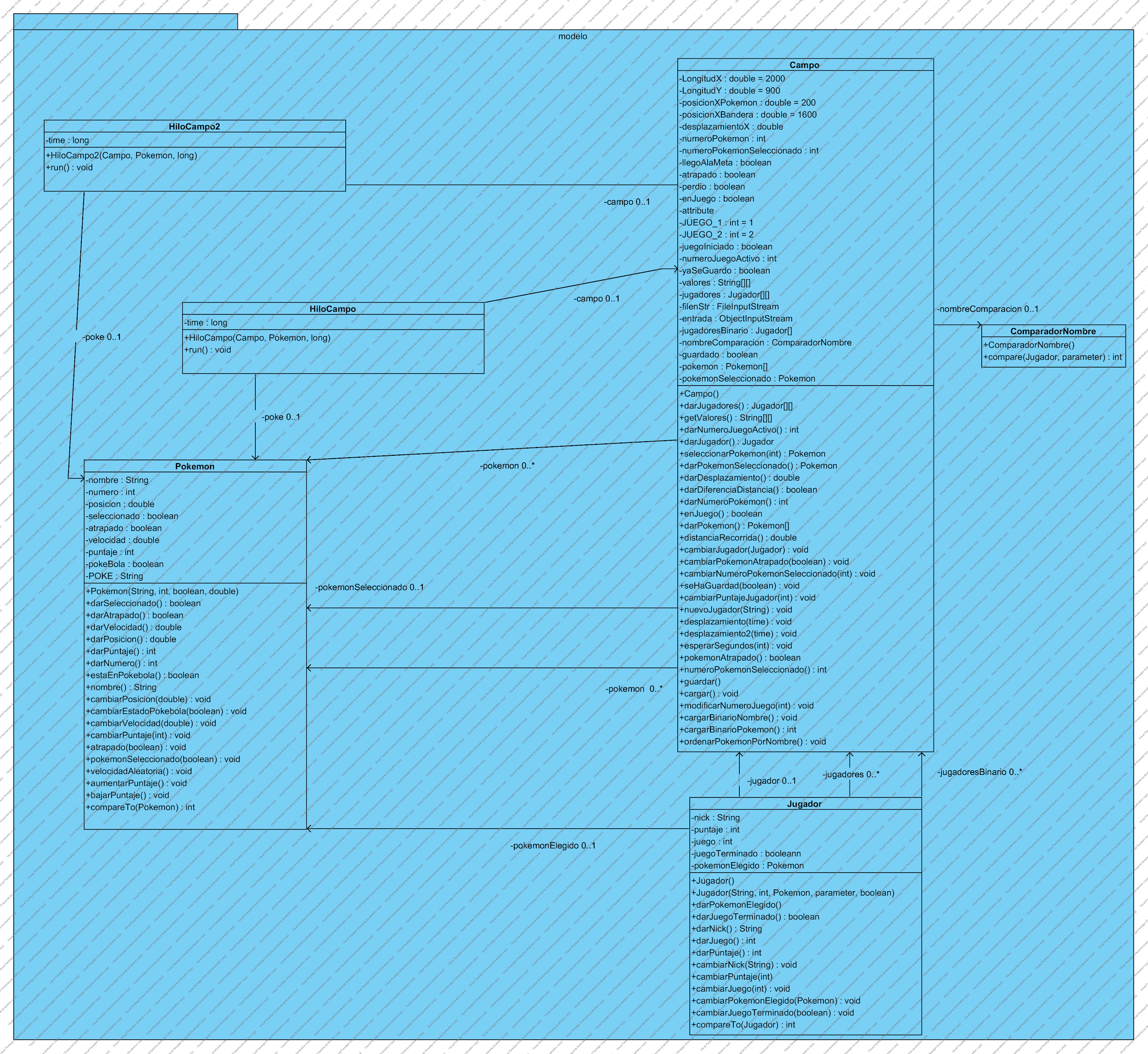
**P2.** Probar que el método guardar() de la clase campo ordena a los jugadores en una lista de acuerdo a su puntaje de manera natural del menor al mayor.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
| Campo  puntajeController | guardar();  initialize(); | nuevoJugador(“abril”);  nuevoJugador(“Carlos”); | “abril” | **Los jugadores han sido ordenados de acuerdo a su orden lexicográfico de manera correcta.** |

**4. Diagrama de pruebas automáticas**

****

**4. Diagrama de clases del modelo:**

****

**5. Diagrama de clases de la interfaz:**

