Diseño del sistema

**IType**

Aplicación Android para

pruebas de velocidad de tipeo

**Versión 1.0**

**Presentado por:**

Cancinos, José María

Oliva, Nahuel David

**Fecha de la Versión:** 14/06/2019

**Índice**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **-** | **HISTORIAL DE VERSIONES** | |  | **3** |
| **1** | **INTRODUCCIÓN** | |  | **4** |
|  | 1.1 | Propósito del documento | ………………. | 4 |
|  | 1.2 | Convenciones del documento | ………………. | 4 |
|  | 1.3 | Destinatarios del documento | ………………. | 4 |
|  | 1.4 | Referencias | ………………. | 4 |
| **2** | **DIAGRAMAS** | |  | **5** |
|  | 2.1 | Clase | ………………. | 6 |
|  | 2.2 | Objetos | ………………. | 8 |
|  | 2.3 | Estado | ………………. | 9 |
|  | 2.4 | Secuencia | ………………. | 10 |
| **3** | **PATRONES DE DISEÑO** | |  | **14** |
|  | 3.1 | Stategy | ………………. | 14 |
|  | 3.2 | Observer | ………………. | 16 |
|  | 3.3 | Singleton | ………………. | 19 |
| **4** | **PRUEBAS UNITARIAS** | |  |  |
|  | 4.1 | Pruebas unitarias aplicadas | ………………. | 21 |
|  | 4.2 | Actualización de matriz de trazabilidad | ………………. | 29 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Historial de versiones**

|  |  |  |
| --- | --- | --- |
| **Nombre de la versión** | **Fecha** | **Número de versión** |
| IType – Diseño del sistema | 14/06/2019 | 1.0 |
|  |  |  |
|  |  |  |

1. **Introducción**
   1. **Propósito del documento**

El presente documento está destinado a exponer cómo se han implementado las decisiones en cuanto al diseño del sistema en un programa, código. Presentar las relaciones de los componentes de software y sus relaciones.

Para ello, se han decidido y precisado aplicar ciertos patrones de diseño de software. Éstos y los problemas que han sido solucionado por medio de dicho uso de patrones se explicitará en el desarrollo de este informe.

* 1. **Convenciones del documento**

Este documento ha sido creado en base a los requerimientos establecidos en la consigna del correspondiente trabajo práctico brindado por la cátedra.

* 1. **Destinatarios del documento**

Este documento está destinado a los usuarios y a potenciales desarrolladores. Así como también toda persona con interés en el proyecto que se lleva a cabo.

* 1. **Referencias**

|  |  |
| --- | --- |
| ELEMENTO | REFERENCIA O DIRECCIÓN WEB |
| Filminas de la materia | https://sites.google.com/view/ingenieria-software-unc/ingeniería-de-software |
| Material de lectura 1 | “Head First: Design Patterns” K. Sierra – E. Freeman |
| Material de lectura 2 | “Ingeniería de Software” I. Sommerville |
|  |  |
|  |  |

1. **Diagramas**

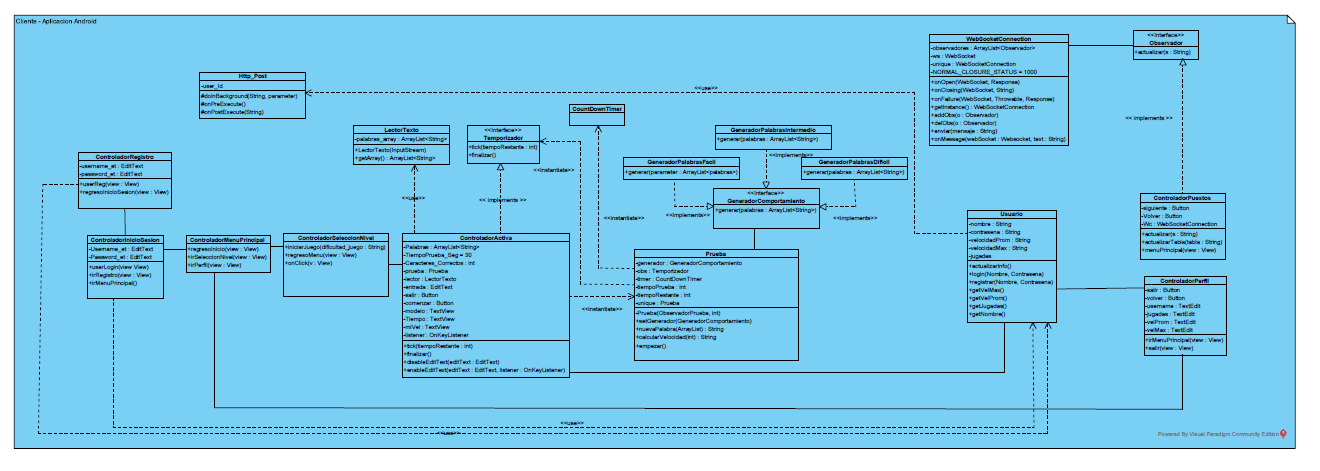
NOTA: Se adjuntan las imágenes de los diagramas en baja calidad para mostrar la existencia de las mismas. Por favor, observarlas a partir del archivo “.pdf” correspondiente presente en la carpeta del archivo de entrega de los informes.

* 1. **Diagrama de clase**

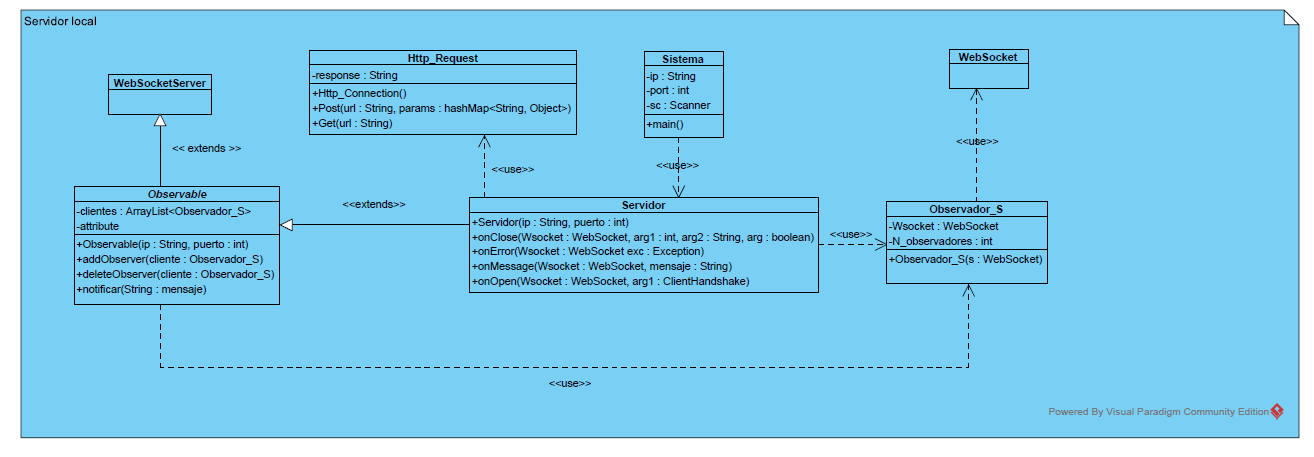
Responde a un modelo estructural al que se arriba en la definición del diseño del sistema.

Luego de distintas iteraciones del proceso de desarrollo, se ha arribado a la siguiente distribución del sistema modelado con lenguaje orientado a objetos. Se pueden conocer las diferentes clases y sus relaciones con las restantes que conforman el sistema. Así también el detalle de atributos y operaciones de cada clase. Muestra la relación estática del sistema.

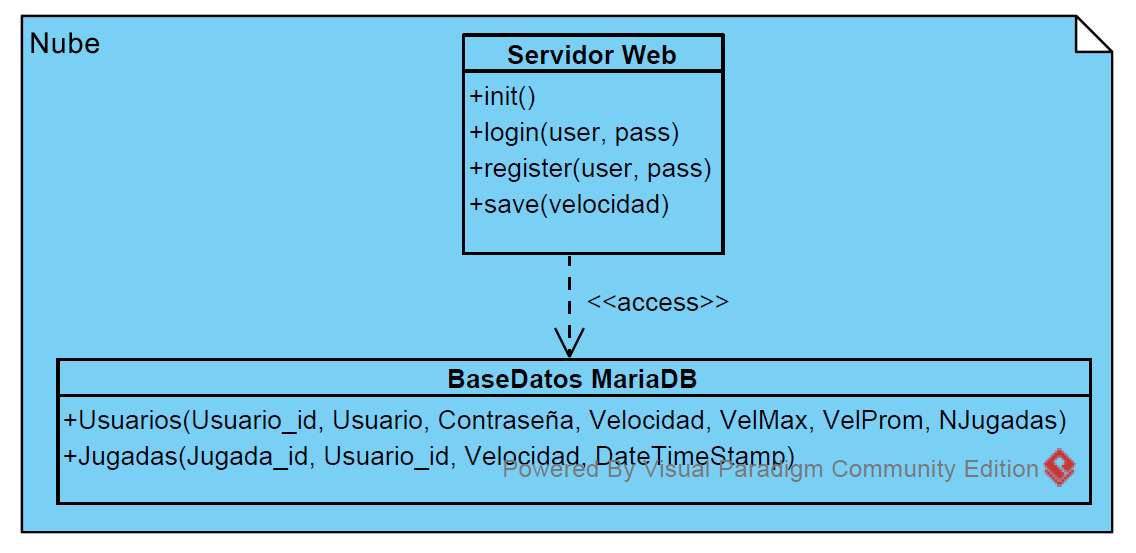
Aplicación Android:



Servidor local

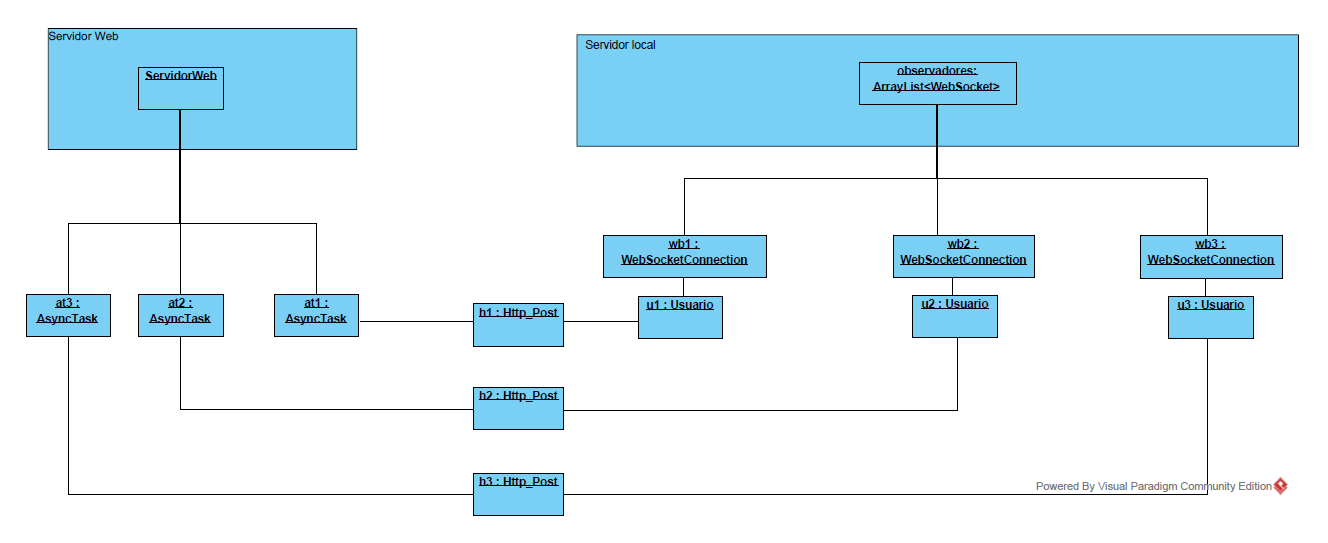


Servidor web



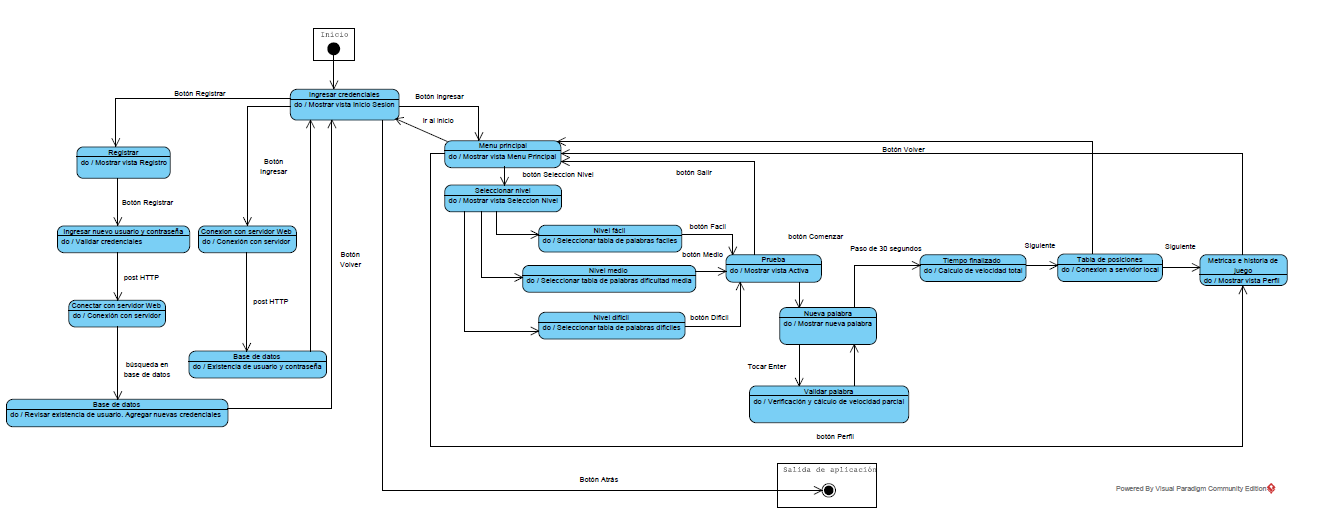
* 1. **Diagramas de objetos**

Se presenta el diagrama de objetos para conocer la relación entre las diferentes instancias que se encuentran en un instante de tiempo de ejecución del programa. Esto permite visualizar la cantidad de objetos posibles de una clase al correr el programa.



* 1. **Diagrama de estado**

Nos permite observar como los objetos cambian de estados dependiendo de los estímulos que reciben.

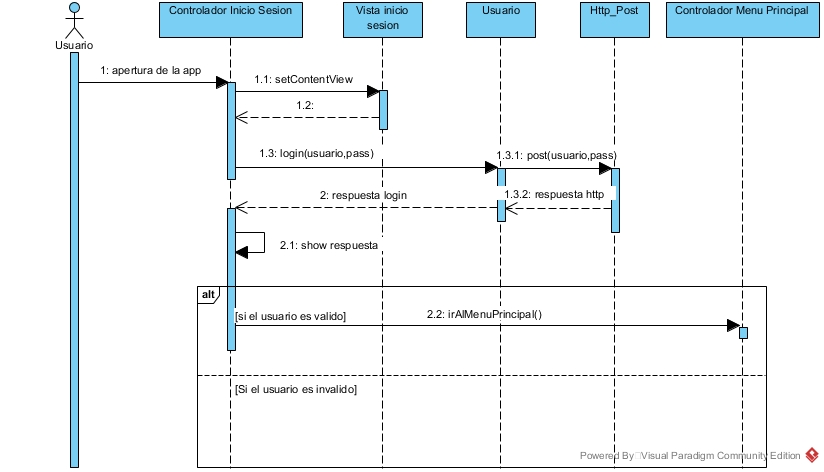


* 1. **Diagramas de secuencia**

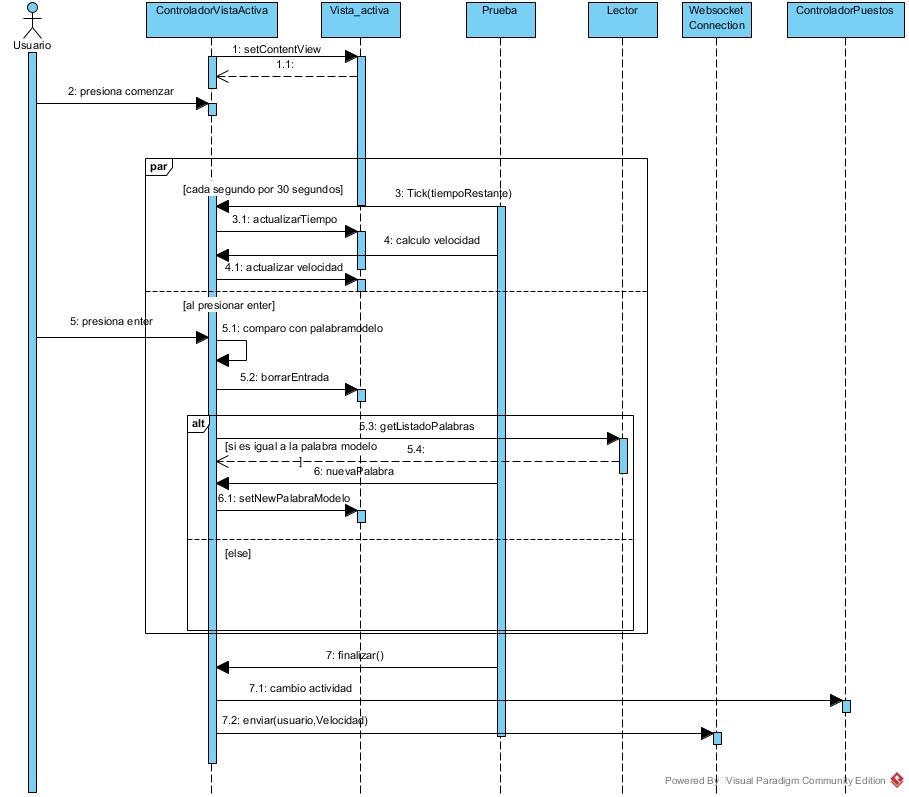
Se puede observar el comportamiento de los objetos frente a distintos mensajes o acciones que suceden.

Nos es útil en la implementación para conocer cómo responden los procesos y el orden de ocurrencia de los mismos en tiempo de ejecución.

Inicio sesión



Prueba (juego)



Vista Puestos y Perfil

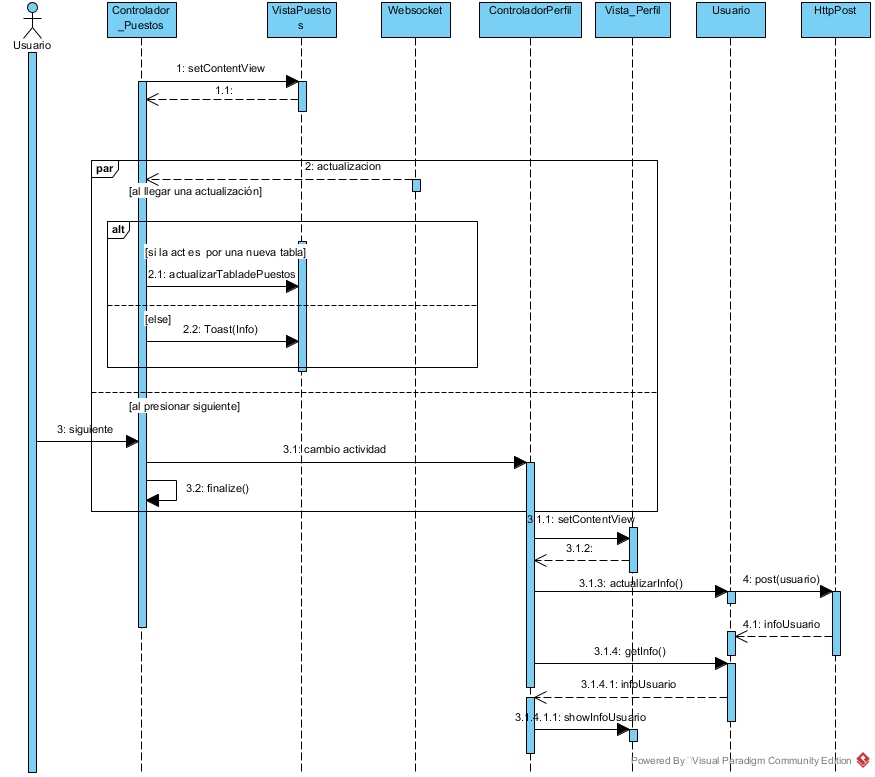
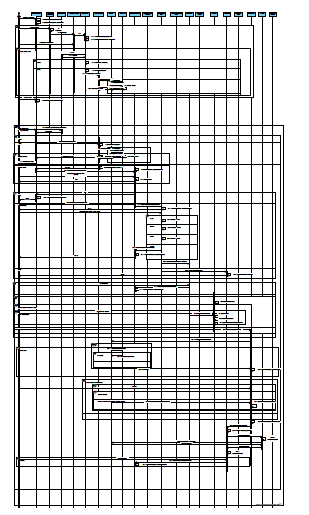


Diagrama completo



1. **Patrones de diseño**

Los patrones de diseño que se han aplicado en el código del proyecto son:

1. Patrón “Strategy”
2. Patrón “Observer”
3. Patrón “Singleton”
   1. **Patrón Strategy**

3.1.1) Explicación del patrón

El patrón Strategy permite que pueda cambiarse el comportamiento de un algoritmo en tiempo de ejecución.

Las características son:

* Define una familia de algoritmos
* Encapsula cada algoritmo
* Permite el intercambio de dichos algoritmos

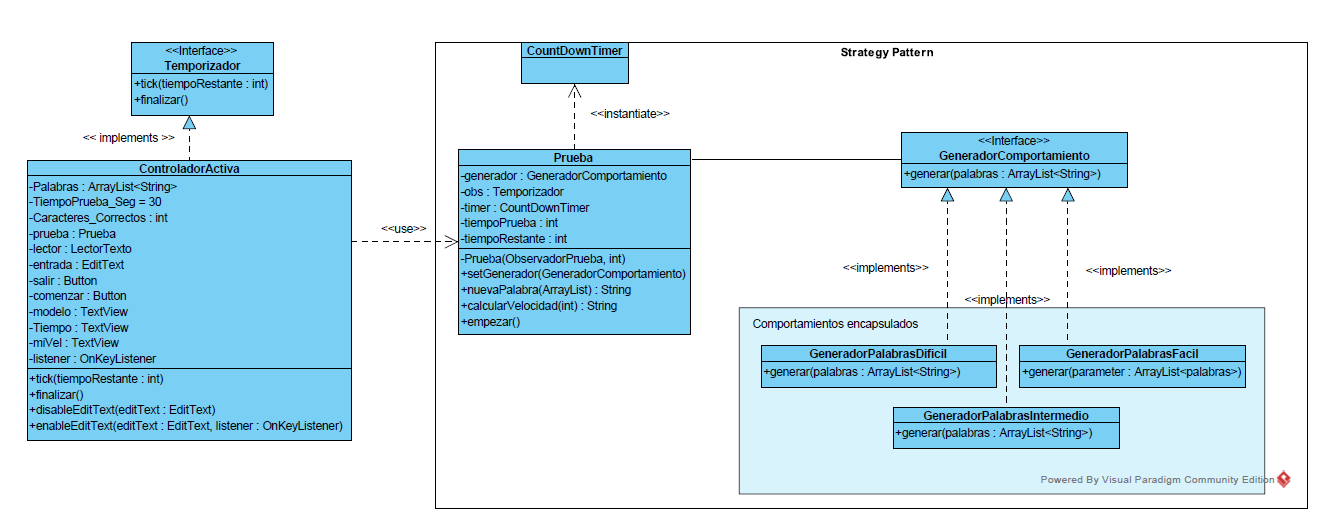
En el proyecto, se emplea el patrón Strategy para implementar los distintos niveles de dificultad del modo de juego.

El método generar() se encapsula para distintas implementaciones dependiendo el nivel: Fácil, Intermedio y Difícil. Este algoritmo está definido en la interfaz GeneradorComportamiento y ésta interfaz es implementada con dicho algoritmo por GeneradorPalabrasFacil, GeneradorPalabrasIntermedio y GeneradorPalabrasDificil.

Cada una de estas clases sobreescribe el método generar() con su propia implementación para obtener una lista de palabras asociadas al nivel correspondiente.

Esto se puede hacer en tiempo de ejecución, es decir, con el programa en funcionamiento e intercambiar entre los distintos comportamientos.

3.1.2) Diagramas de clase asociados al patrón



3.1.3) Solución de problemas mediante aplicación del patrón

El patrón Strategy nos permite evitar el uso de estructuras de decisión como “if-else” o swtich para elegir entre las dificultades. Además, permite separar en distintas clases la implementación para cada dificultad. De otra forma, hubiéramos tenido que definir el código para los tres casos en una misma clase y no se lograría la modularización óptima.

Strategy permite el cambio de dificultad en tiempo de ejecución. Es decir, podemos cambiar la dificultad al terminar una prueba sin tener que salir de la aplicación o por definición previa.

Además, como se ha desarrollado la interfaz GeneradorComportamiento, es posible agregar más niveles en un futuro y sólo deberíamos hacer que el nuevo nivel implemente dicha interfaz.

* 1. **Patrón Observer**

3.2.1) Explicación del patrón

El patrón Observer define dos tipos de participantes:

* El sujeto o “el observado”: aquel que contiene información útil para los observadores
* Los observadores: aquellos que se encuentran expectantes frente a cambio de datos del sujeto y reflejan un estado del mismo.

Observer permite que varios observadores se definan como tal frente al sujeto. De tal forma que cuando el sujeto cambie su información, los observadores puedan ser notificados y automáticamente reflejar los cambios.

La característica es que es el sujeto quien coloca los datos a disposición de los observadores.

Existen clases abstractas de sujetos y observadores.

En nuestro caso, la clase abstracta de sujeto es “Observable” y la clase abstracta de los observadores es “Observador”

WebSocketConnection actúa como intermediario entre el servidor y la aplicación.

Para las clases concretas, el sujeto es el Servidor. Éste contiene la información necesaria para desplegar la tabla de puestos de jugadas globales.

Los observadores, en este caso, son las instancias de la clase ControladorPuestos de cada usuario de la aplicación. Éstos deben poder reflejar la tabla de posiciones actualizada en todo momento.

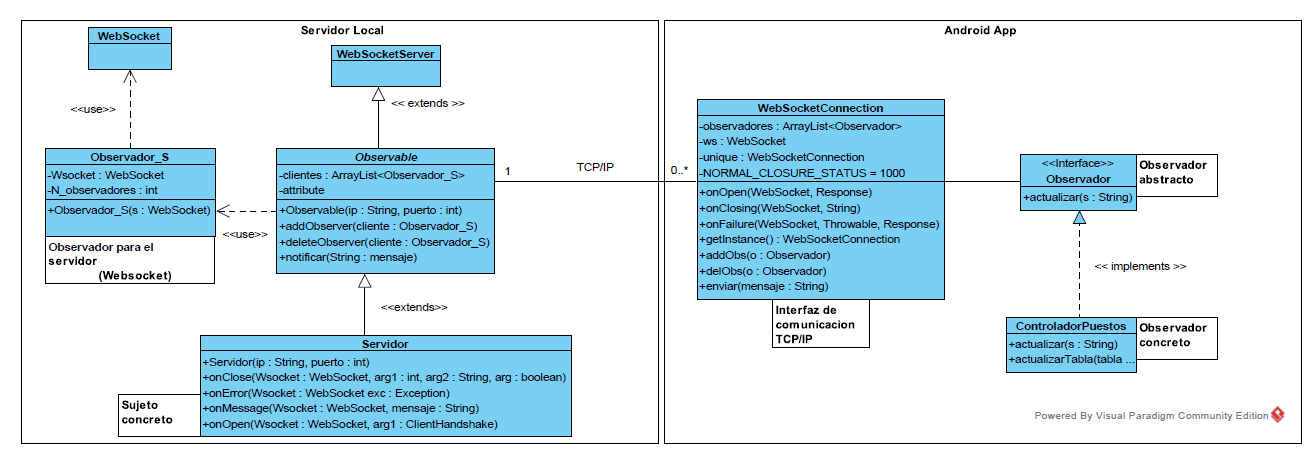
También encontramos observadores para el propio servidor en “Observador\_S”

La aplicación de Observer en este caso es tal que cuando la clase sujeto concreto Servidor cambie de estado, la clase abstracta Observable implemente el notificar() hacia observadores. Las clases ControladoresPuestos en cada instancia de teléfono ejecutará el actualizar() para recibir el nuevo estado del servidor y poder reflejarlo.

En el lado del servidor, el sujeto concreto sigue siendo Servidor y el sujeto abstracto continúa siendo Observable mientras que los observadores concretos son WebSocket y observadores abstractos Observador\_S. En este caso, es necesario reflejar la cantidad de clientes de la aplicación en linea. Esto se logra mediante Observer también.

Todo nuevo cliente pasa a formar parte de la lista suscripta en Observable y puede ser eliminado también.

3.2.2) Diagramas de clase asociados al patrón



3.2.3) Solución de problemas mediante aplicación del patrón

Observer nos garantiza que cada usuario que quiera adquirir los datos del servidor, se agregue a la lista y también pueda salir de manera fácil y con tan sólo ejecutar un método.

Así también se produce con la actualización. Los observadores no tienen que preguntar si los datos del servidor han cambiado y hay nuevos valores de la tabla. Es el propio Servidor que ejecuta un método notificar() hacia los observadores suscriptos y éstos reciben los datos. ´Caso contrario, el realizar “pooling” implica tiempo y recursos para evaluar el estado.

Es muy fácil poder agregar o quitar observadores. No es necesario el cambio de la estructura del sujeto para poder realizar estas operaciones.

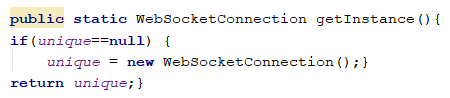
* 1. **Patrón Singleton**

3.3.1) Explicación del patrón

El patrón Singleton permite limitar a la existencia de una única instancia de clase. No puede haber más de una.

WebSocketConnection:





Se observa la aplicación del patrón Singleton para obtener solamente una conexión al servidor local.

La instancia de la clase a crear es definida como estática, de tal forma, que es una variable de clase.

Al momento de crear el objeto, se llama al método estático getInstance(). Se comprueba que el objeto “unique” exista o no. Si no existe, éste se crea. Si ya existe, se entrega el ya creado.

3.3.2) Diagramas de clase asociados al patrón

Se muestra un diagrama de objetos para mostrar tres instancias de la aplicación. Se observa que cada una de ellas contiene un único y propio objeto WebSocketConnection.

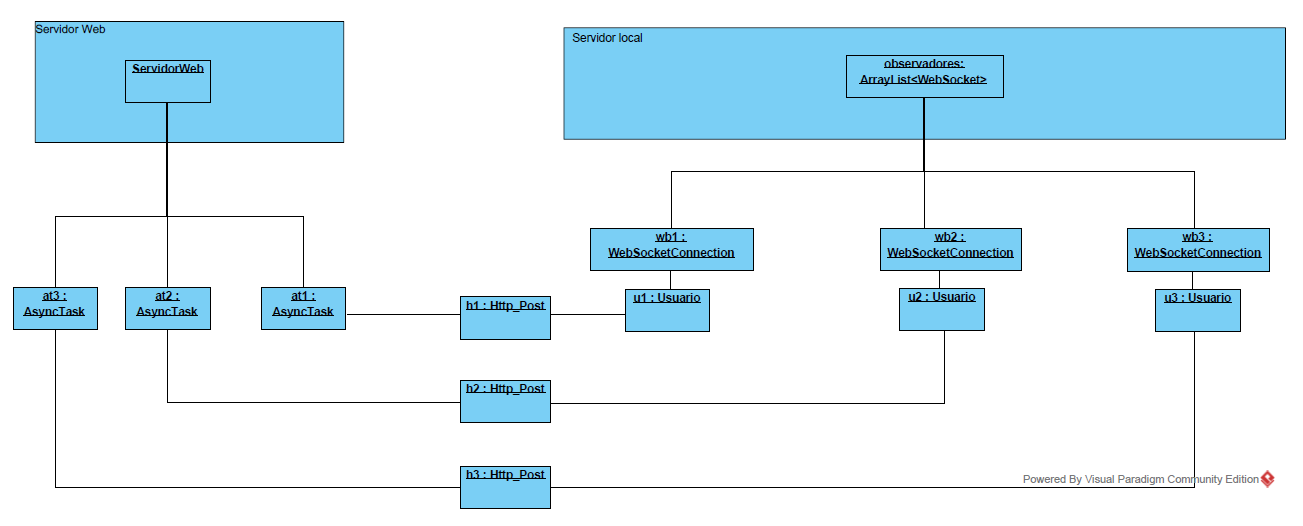
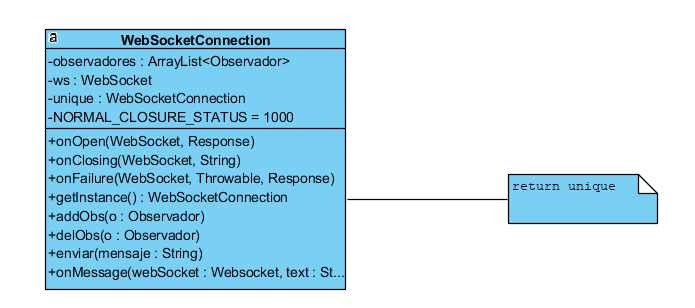


Diagrama de clase - Singleton



3.3.3) Solución de problemas mediante aplicación del patrón

Singleton nos asegura que solamente existirá una única instancia de la clase WebSocketConnection. Por lo tanto, existirá una única conexión TCP/IP con el servidor local. Esto es lo requerido.

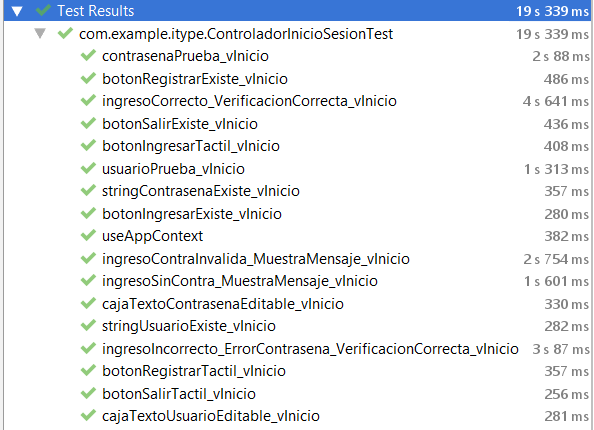
En caso de haber dos o más instancias con el mismo usuario, esto crearía problemas en la actualización de datos ya que actualizaciones pueden correr a diferente tiempo si tenemos más de una instancia. Además, es un único usuario quién realiza el juego, por lo tanto, esto debe ser representado.

También, es necesario que exista una única instancia de conexión al servidor para garantizar la seguridad. No puede un mismo usuario correr acceder, registrarse o modificar datos en el servidor con múltiples instancias de él mismo.

1. **Pruebas unitarias**
   1. **Pruebas unitarias aplicadas**

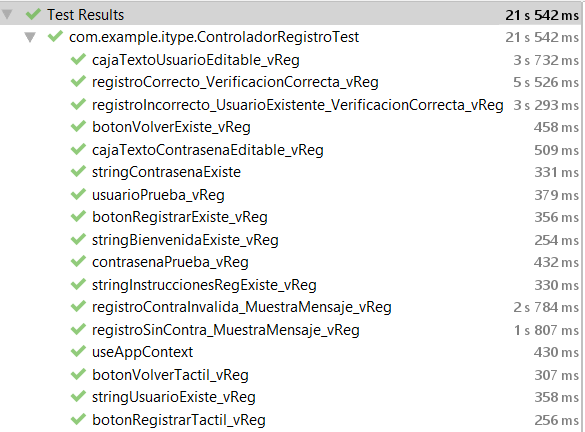
En esta sección se especifican las pruebas unitarias realizadas para el “testeo” del sistema. Algunas se han llevado a cabo mediante JUnit mientras que la gran mayoría mediante Espresso/Junit.

Clase ControladorInicioSesion



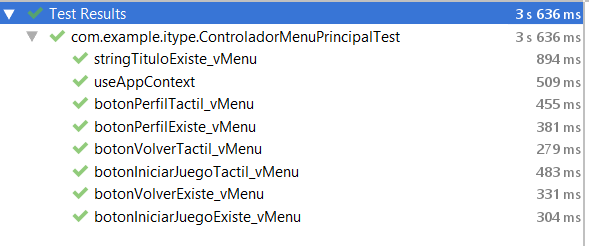
|  |  |  |
| --- | --- | --- |
| Pruebas unitarias | | |
| Id | Nombre | Descripción |
| PU1.1 | useAppContext() | Verifico que la actividad actual forma parte del paquete “com.example.itype”; paquete de nuestra aplicación. |
| PU1.2 | stringUsuarioExiste\_vInicio() | Verificación de existencia y presentación de texto “Usuario” en la vista. |
| PU1.3 | stringContrasenaExiste\_vInicio() | Verificación de existencia y presentación de texto “Contraseña” en la vista. |
| PU1.4 | botonIngresarExiste\_vInicio() | Verificación de existencia y presentación del botón Ingresar. |
| PU1.5 | botonRegistrarExiste\_vInicio() | Verificación de existencia y presentación del botón Registrar. |
| PU1.6 | botonIngresarTactil\_vInicio() | Verificación de posibilidad de hacer clic en botón Ingresar. |
| PU1.7 | botonRegistrarTactil\_vInicio() | Verificación de posibilidad de hacer clic en botón Registrar. |
| PU1.8 | cajaTextoUsuarioEditable\_vInicio() | Verificación de posibilidad de relleno por usuario para nombre de usuario |
| PU1.9 | cajaTextoContrasenaEditable\_vInicio() | Verificación de posibilidad de relleno por usuario para contraseña |
| PU1.10 | usuarioPrueba\_vInicio() | Verifico que lo que escribí en la caja de texto de usuario es igual a lo desplegado |
| PU1.11 | contrasenaPrueba\_vInicio() | Verifico que lo que escribí en la caja de texto de contraseña es igual a lo desplegado |
| PU1.12 | ingresoCorrecto\_  VerificacionCorrecta\_vInicio() | Ingreso usuario y contraseña, hago clic en Ingresar y verifico que el mensaje de bienvenida se despliega |
| PU1.13 | ingresoSinContra\_  MuestraMensaje\_vInicio() | Ingreso usuario pero contraseña vacía, hago clic en Ingresar y verifico que el mensaje de error se despliega |
| PU1.14 | ingresoContraInvalida\_  MuestraMensaje\_vInicio() | Ingreso usuario, pero contraseña con carácter inválido. Verifico muestra de mensaje de error |
| PU1.15 | ingresoIncorrecto\_ErrorContrasena\_  VerificacionCorrecta\_vInicio() | Ingreso usuario existente, pero con diferente contraseña. Verifico despliegue de mensaje de credenciales incorrectos |
| PU1.16 | botonSalirExiste\_vInicio() | Verificación de existencia y presentación del botón Salir. |
| PU1.17 | botonSalirTactil\_vInicio() | Verificación de posibilidad de hacer clic en botón Salir. |

Clase ControladorRegistro



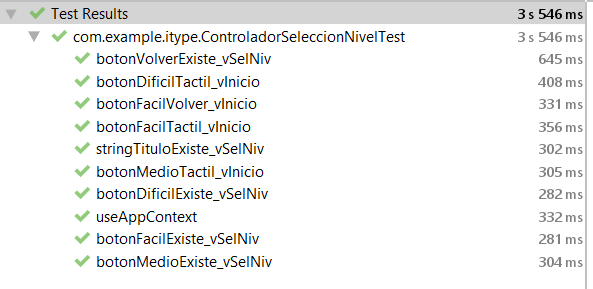
|  |  |  |
| --- | --- | --- |
| Pruebas unitarias | | |
| Id | Nombre | Descripción |
| PU2.1 | useAppContext() | Verifico que la actividad actual forma parte del paquete “com.example.itype”, |
| PU2.2 | stringUsuarioExiste\_vReg() | Verificación de existencia y presentación de texto “Usuario” en la vista. |
| PU2.3 | stringContrasenaExiste() | Verificación de existencia y presentación de texto “Contraseña” en la vista. |
| PU2.4 | stringBienvenidaExiste\_vReg() | Verificación de existencia y presentación de texto de bienvenida nuevo usuario en la vista. |
| PU2.5 | stringInstruccionesRegExiste\_vReg() | Verificación de existencia y presentación de texto de instrucciones para registro en la vista. |
| PU2.6 | botonVolverExiste\_vReg() | Verificación de existencia y presentación del botón Volver. |
| PU2.7 | botonRegistrarExiste\_vReg() | Verificación de existencia y presentación del botón Registrar. |
| PU2.8 | botonVolverTactil\_vReg() | Verificación de posibilidad de hacer clic en botón Volver. |
| PU2.9 | botonRegistrarTactil\_vReg() | Verificación de posibilidad de hacer clic en botón Registrar. |
| PU2.10 | cajaTextoUsuarioEditable\_vReg() | Verificación de posibilidad de relleno por usuario para nombre de usuario |
| PU2.11 | cajaTextoContrasenaEditable\_vReg() | Verificación de posibilidad de relleno por usuario para contraseña de usuario |
| PU2.12 | contrasenaPrueba\_vReg() | Verifico que lo que escribí en la caja de texto de contraseña es igual a lo desplegado |
| PU2.13 | usuarioPrueba\_vReg() | Verifico que lo que escribí en la caja de texto de usuario es igual a lo desplegado |
| PU2.14 | registroCorrecto  \_VerificacionCorrecta\_vReg() | Ingreso usuario y contraseña nuevo. Verifico despliegue de nuevo usuario correcto |
| PU2.15 | registroSinContra  \_MuestraMensaje\_vReg() | Ingreso usuario y contraseña vacía. Verifico despliegue de error al registrar |
| PU2.16 | registroContraInvalida  \_MuestraMensaje\_vReg() | Ingreso usuario y contraseña con carácter no permitido. Verifico despliegue de error al registrar |
| PU2.17 | registroIncorrecto\_UsuarioExistente  \_VerificacionCorrecta\_vReg() | Ingreso usuario y contraseña ya existente. Verifico despliegue de usuario duplicado |

Clase ControladorMenuPrincipal

****

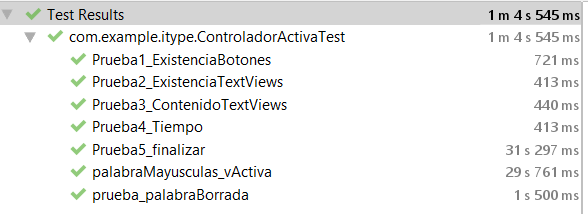
|  |  |  |
| --- | --- | --- |
| Pruebas unitarias | | |
| Id | Nombre | Descripción |
| PU3.1 | useAppContext() | Verifico que la actividad actual forma parte del paquete “com.example.itype”, |
| PU3.2 | stringTituloExiste\_vMenu() | Verificación de existencia y presentación de texto “MENÚ PRINCIPAL” en la vista. |
| PU3.3 | botonIniciarJuegoExiste\_vMenu() | Verificación de existencia y presentación del botón Iniciar Juego. |
| PU3.4 | botonPerfilExiste\_vMenu() | Verificación de existencia y presentación del botón Perfil. |
| PU3.5 | botonVolverExiste\_vMenu() | Verificación de existencia y presentación del botón Volver. |
| PU3.6 | botonIniciarJuegoTactil\_vMenu() | Verificación de posibilidad de hacer clic en botón Inicio Juego. |
| PU3.7 | botonPerfilTactil\_vMenu() | Verificación de posibilidad de hacer clic en botón Perfil. |
| PU3.8 | botonVolverTactil\_vMenu() | Verificación de posibilidad de hacer clic en botón Volver. |

Clase ControladorSeleccionNivel

****

|  |  |  |
| --- | --- | --- |
| Pruebas unitarias | | |
| Id | Nombre | Descripción |
| PU4.1 | useAppContext() | Verifico que la actividad actual forma parte del paquete “com.example.itype”, |
| PU4.2 | stringTituloExiste\_vSelNiv( | Verificación de existencia y presentación de texto “Seleccione la dificultad del juego” en la vista. |
| PU4.3 | botonFacilExiste\_vSelNiv() | Verificación de existencia y presentación de botón “Fácil” en la vista. |
| PU4.4 | botonMedioExiste\_vSelNiv() | Verificación de existencia y presentación de botón “Media” en la vista. |
| PU4.5 | botonDificilExiste\_vSelNiv() | Verificación de existencia y presentación de botón “Difícil” en la vista. |
| PU4.6 | botonVolverExiste\_vSelNiv() | Verificación de existencia y presentación de botón “Fácil” en la vista. |
| PU4.7 | botonFacilTactil\_vInicio() | Verificación de posibilidad de hacer clic en botón Fácil |
| PU4.8 | botonMedioTactil\_vInicio() | Verificación de posibilidad de hacer clic en botón Medio |
| PU4.9 | botonDificilTactil\_vInicio() | Verificación de posibilidad de hacer clic en botón Difícil |
| PU4.10 | botonFacilVolver\_vInicio() | Verificación de posibilidad de hacer clic en botón Volver |

Clase ControladorActiva



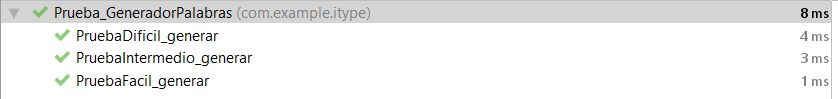
|  |  |  |
| --- | --- | --- |
| Pruebas unitarias | | |
| Id | Nombre | Descripción |
| PU5.1 | Prueba1\_ExistenciaBotones() | Verifico existencia y despliegue de botón Comenzar y botón Menú Principal |
| PU5.2 | Prueba2\_ExistenciaTextViews() | Pruebo existencia y despliegue de texto “ Tiempo”, “Velocidad”, “Palabra modelo” |
| PU5.3 | Prueba3\_ContenidoTextViews() | Verifico que los parámetros de juego son correctamente desplegados y coinciden. |
| PU5.4 | Prueba4\_Tiempo() | Verifico que el tiempo comienza a desplegarse |
| PU5.5 | Prueba5\_finalizar( | Verifico que al final de la prueba vacía el puntaje el tiempo es 0 y se cambia automáticamente la vista. |
| PU5.6 | palabraMayusculas\_vActiva() | Verifico que las palabras serán reconocidas igual si están en mayúsculas |
| PU5.7 | prueba\_palabraBorrada() | Verifico que al completar una palabra y hacer clic en Enter, el cajón de texto se borra |

Clase Lector\_textoTest



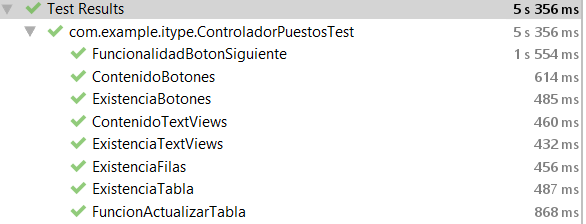
|  |  |  |
| --- | --- | --- |
| Pruebas unitarias | | |
| Id | Nombre | Descripción |
| PU6.1 | Prueba1\_Creacion\_Lector() | Pruebo la correcta creación de un objeto Lector |
| PU6.2 | Prueba2\_LecturaPalabras() | Pruebo que la obtención de la lista de palabras es correcta |

Clase Prueba\_GeneradorPalabras



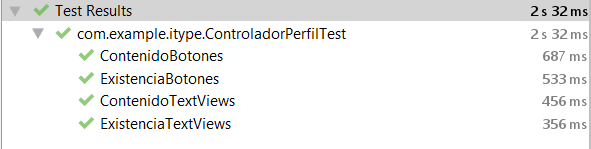
|  |  |  |
| --- | --- | --- |
| Pruebas unitarias | | |
| Id | Nombre | Descripción |
| PU7.1 | PruebaFacil\_generar() | Evalúo la obtención desde el texto hacia el arreglo solamente de palabras con nivel fácil |
| PU7.2 | PruebaIntermedio\_generar() | Evalúo la obtención desde el texto hacia el arreglo solamente de palabras con nivel medio |
| PU7.3 | PruebaDificil\_generar() | Evalúo la obtención desde el texto hacia el arreglo solamente de palabras con nivel difícil |

Clase ControladorPuestos



|  |  |  |
| --- | --- | --- |
| Pruebas unitarias | | |
| Id | Nombre | Descripción |
| PU8.1 | ExistenciaTextViews() | Verifico que los textos de la tabla en columnas “Usuario”, “Puesto” y “Velocidad” existe y se despliega |
| PU8.2 | ExistenciaBotones() | Verifico existencia y despliegue de botón Siguiente y Menú Principal |
| PU8.3 | ExistenciaTabla() | Verifico existencia y despliegue de la tabla |
| PU8.4 | ExistenciaFilas() | Verifico existencia y despliegue de 10 filas de la tabla más una de encabezado |
| PU8.5 | ContenidoTextViews() | Verifico el correcto despliegue del texto en el encabezado de tabla |
| PU8.6 | ContenidoBotones() | Verifico que los botones desplieguen el texto correcto |
| PU8.7 | FuncionActualizarTabla() | Verifico que los datos se agregan y despliegan para todos los usuarios. |
| PU8.8 | FuncionalidadBotonSiguiente() | Pruebo el fin de actividad actual cuando se hace clic en botón Siguiente |

Clase ControladorPerfil



|  |  |  |
| --- | --- | --- |
| Pruebas unitarias | | |
| Id | Nombre | Descripción |
| PU9.1 | ExistenciaTextViews() | Evalúo la existencia de texto desplegado “Usuario”; “Velocidad Promedio”, “Jugadas” y “Velocidad Máxima” |
| PU9.2 | ExistenciaBotones() | Pruebo la existencia y despliegue del botón Volver |
| PU9.3 | ContenidoTextViews() | Verifico que el texto desplegado coincida con el deseado |
| PU9.4 | ContenidoBotones() | Evalúo que el texto desplegado en el botón sea “Menú Principal” |

* 1. **Actualización de matriz de trazabilidad**

Relación entre requerimientos funcionales y no funcionales contra pruebas unitarias

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pruebas unitarias / Req | PU1.1 | PU1.2 | PU1.3 | PU1.4 | PU1.5 | PU1.6 | PU1.7 | PU1.8 | PU1.9 | PU1.10 | PU1.11 | PU1.12 | PU1.13 | PU1.14 | PU1.15 | PU1.16 | PU1.17 | PU2.1 | PU2.2 | PU2.3 | PU2.4 | PU2.5 | PU2.6 | PU2.7 | PU2.8 | PU2.9 | PU2.10 | PU2.11 | PU2.12 | PU2.13 | PU2.14 | PU2.15 | PU2.16 | PU2.17 |
| R1.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R1.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R1.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R1.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.1 | X | X | X | X | X |  |  | X | X |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R3.1 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| R3.2 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| R3.3 |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X |
| R4.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN1.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN1.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN1.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN2.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN2.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN2.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN2.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN3.1 |  |  |  |  |  |  |  |  |  |  |  | X | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X | X |
| RN4.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pruebas unitarias / Requerimientos | PU3.1 | PU3.2 | PU3.3 | PU3.4 | PU3.5 | PU3.6 | PU3.7 | PU3.8 | PU4.1 | PU4.2 | PU4.3 | PU4.4 | PU4.5 | PU4.6 | PU4.7 | PU4.8 | PU4.9 | PU4.10 | PU5.1 | PU5.2 | PU5.3 | PU5.4 | PU5.5 | PU5.6 | PU5.7 | PU6.1 | PU6.2 | PU7.1 | PU7.2 | PU7.3 |
| R1.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R1.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |
| R1.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |
| R1.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.1 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |  |  |  |  |  |  |  |  |
| R2.2 |  |  |  |  | X |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |
| R2.6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.7 |  |  |  |  |  |  |  |  |  |  | X | X | X |  | X | X | X |  |  |  |  |  |  |  |  |  |  | X | X | X |
| R3.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R3.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R3.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |
| R4.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |
| R4.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |
| R4.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |
| R4.5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |
| R4.6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN1.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN1.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN1.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN2.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN2.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN2.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN2.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN3.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RN4.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pruebas unitarias / Requerimientos | PU8.1 | PU8.2 | PU8.3 | PU8.4 | PU8.5 | PU8.6 | PU8.7 | PU8.8 | PU9.1 | PU9.2 | PU9.3 | PU9.4 |
| R1.1 |  |  |  |  |  |  |  |  |  |  |  |  |
| R1.2 |  |  |  |  |  |  |  |  |  |  |  |  |
| R1.3 |  |  |  |  |  |  |  |  |  |  |  |  |
| R1.4 |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.1 | X | X | X | X | X | X |  | X | X | X | X | X |
| R2.2 |  | X |  |  |  |  |  |  |  | X |  | X |
| R2.3 |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.4 | X |  | X | X | X |  |  |  |  |  |  |  |
| R2.5 |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.6 |  |  |  | X |  |  |  |  |  |  |  |  |
| R2.7 |  |  |  |  |  |  |  |  |  |  |  |  |
| R3.1 |  |  |  |  |  |  |  |  |  |  |  |  |
| R3.2 |  |  |  |  |  |  |  |  |  |  |  |  |
| R3.3 |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.1 |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.2 |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.3 |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.4 |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.5 |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.6 |  |  |  |  |  |  | X |  |  |  |  |  |
| RN1.1 |  |  |  |  |  |  |  |  |  |  |  |  |
| RN1.2 |  |  |  |  |  |  |  |  |  |  |  |  |
| RN1.3 |  |  |  |  |  |  |  |  |  |  |  |  |
| RN2.1 |  |  |  |  |  |  |  |  |  |  |  |  |
| RN2.2 |  |  |  |  |  |  |  |  |  |  |  |  |
| RN2.3 |  |  |  |  |  |  |  |  |  |  |  |  |
| RN2.4 |  |  |  |  |  |  |  |  |  |  |  |  |
| RN3.1 |  |  |  |  |  |  |  |  |  |  |  |  |
| RN4.1 |  |  |  |  |  |  |  |  |  |  |  |  |

**Nota:**

La matriz de trazabilidad muestra requerimientos sin probar por pruebas unitarias. Éstas serán probadas con pruebas más generales como integración, sistema o de aceptación ya que su alcance es mayor.