Especificación de requerimientos

**IType**

Aplicación Android para

pruebas de velocidad de tipeo

**Versión 4.0**

**Presentado por:**

Cancinos, José María

Oliva, Nahuel David

**Fecha de la Versión:** 14/06/2019

**Índice**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **-** | **HISTORIAL DE VERSIONES** | |  | **4** |
| **1** | **INTRODUCCIÓN** | |  | **5** |
|  | 1.1 | Propósito del documento | ………………. | 5 |
|  | 1.2 | Convenciones del documento | ………………. | 5 |
|  | 1.3 | Destinatarios del documento | ………………. | 5 |
|  | 1.4 | Alcance del producto | ………………. | 5 |
|  | 1.5 | Referencias | ………………. | 6 |
| **2** | **DESCRIPCIÓN GENERAL** | |  | **7** |
|  | 2.1 | Funcionamiento | ………………. | 7 |
|  | 2.2 | Perspectiva del producto | ………………. | 7 |
|  | 2.3 | Funciones del producto | ………………. | 8 |
|  | 2.4 | Clase y características del usuario | ………………. | 8 |
|  | 2.5 | Entorno operativo | ………………. | 9 |
|  | 2.6 | Límites de diseño e implementación | ………………. | 9 |
|  | 2.7 | Documentación de usuario | ………………. | 10 |
|  | 2.8 | Dependencias y suposiciones | ………………. | 10 |
| **3** | **REQUERIMIENTOS DE INTERFAZ EXTERNOS** | |  | **11** |
|  | 3.1 | Interfaces de usuario | ………………. | 11 |
|  | 3.2 | Interfaces de hardware | ………………. | 11 |
|  | 3.3 | Interfaces de software | ………………. | 11 |
|  | 3.4 | Interfaces de comunicación | ………………. | 11 |
| **4** | **REQUERIMIENTOS FUNCIONALES** | |  | **13** |
|  | 4.1 | REQ 1: Entrada del usuario | ………………. | 13 |
|  | 4.2 | REQ 2: Interfaz | ………………. | 14 |
|  | 4.3 | REQ 3: Sesión | ………………. | 15 |
|  | 4.4 | REQ 4: Funcionamiento general | ………………. | 16 |
| **5** | **CASOS DE USO** | |  | **18** |
|  | 5.1 | Definición de casos de uso | ………………. | 18 |
|  | 5.2 | Diagrama de casos de uso | ………………. | 19 |
|  | 5.3 | Diagrama de actividades | ………………. | 20 |
|  | 5.4 | Matriz de trazabilidad | ………………. | 21 |
| **6** | **REQUERIMIENTOS NO FUNCIONALES** | |  | **22** |
|  | 6.1 | Requerimientos de rendimiento | ………………. | 22 |
|  | 6.2 | Requerimientos de eficiencia | ………………. | 22 |
|  | 6.3 | Requerimientos de seguridad | ………………. | 22 |
|  | 6.4 | Atributos de calidad del software | ………………. | 22 |
| **7** | **CASOS DE PRUEBA DE SISTEMA** | |  | **23** |
|  | 7.1 | Definición de casos de prueba de sistema | ………………. | 24 |
|  | 7.2 | Definición de casos de prueba alternativos | ………………. | 30 |
|  | 7.3 | Pruebas de aceptación | ………………. | 31 |
|  | 7.4 | Pruebas smoke | ………………. | 31 |
|  | 7.5 | Matriz de trazabilidad | ………………. | 32 |
| **8** | **DIAGRAMAS ADICIONALES** | |  | **34** |
|  | 8.1 | Diagrama de secuencia para registro | ………………. | 34 |
|  | 8.2 | Diagrama de secuencia para juego | ………………. | 35 |
| **9** | **ARQUITECTURA PRELIMINAR** | |  | **36** |
|  | 9.1 | Diagrama de arquitectura preliminar | ………………. | 36 |

**Historial de versiones**

|  |  |  |
| --- | --- | --- |
| **Correcciones** | **Fecha** | **Número de versión** |
| Versión Inicial | 10/04/2019 | 1.0 |
| Agregado de nuevos requerimientos de sistema | 21/04/2019 | 2.0 |
| Correcciones de requerimientos funcionales | 2/05/2019 | 3.0 |
| Correcciones de casos de prueba de sistema y diagramas | 4/06/2019 | 4.0 |

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

El propósito del siguiente documento es de presentar una descripción detallada sobre el software **IType**.

Se explicará el propósito, las características, las interfaces, la funcionalidad y los límites en que se deberá operar.

* 1. **Convenciones del documento**

Este documento ha sido creado en base a la planilla IEEE[[1]](#footnote-1) para documentos de especificación de requerimientos.

* 1. **Destinatarios del documento**

Este documento está destinado a los usuarios y a potenciales desarrolladores.

* 1. **Alcance del producto**

IType es una aplicación destinada al sistema operativo Android, en la cual se pone a prueba la velocidad de “tipeo” del usuario. El objetivo es escribir la mayor cantidad de caracteres de manera correcta durante el tiempo de duración de la prueba. Luego, al terminar la prueba, se da a conocer la velocidad de escritura que se obtuvo y si se logró calificar en la tabla de posiciones.

La aplicación, también, ofrece la posibilidad de escoger distintos niveles de dificultad, permitiendo así el uso tanto a usuarios expertos como novatos.

* 1. **Referencias**

|  |  |
| --- | --- |
| ELEMENTO | REFERENCIA O DIRECCIÓN WEB |
| Repositorio del proyecto en GitLab: | <https://gitlab.com/Nadaol/typing_app> |
| Planilla de SRS – Modelo IEEE | <https://goo.gl/nsUFwy> |
| Licencia GNU - Versión 3.0 | <https://opensource.org/licenses/GPL-3.0> |
| Base de datos - MySQL |  |
| Documentación de Java | <https://docs.oracle.com/javase/7/docs/api/index-files/index-1.html> |

**2. Descripción General**

1. **Funcionamiento**

La aplicación IType constará de una interfaz con siete vistas interactivas:

1. La vista de **inicio de sesión** será cargada al iniciar la aplicación. Ésta permitirá al usuario iniciar sesión con una cuenta ya registrada para poder guardar la información de partida a la base de datos.
2. La vista de **registro** de usuario la cual permitirá la creación de una nueva cuenta.
3. La vista de **menú principal**, la cual permite empezar la prueba, visitar la vista perfil o regresar al inicio de sesión.
4. En la vista de **selección de nivel** se podrán elegir tres modos de dificultad para la prueba:

* Fácil.
* Intermedio.
* Difícil.

1. La vista **activa** es aquella en la que se corre la prueba en sí, luego de haber elegido la dificultad de la misma; mostrando la palabra a “tipear” y permitiendo la entrada al usuario.
2. Luego, una vez que la prueba finaliza, se podrá cargar la vista de **puestos**. Ésta muestra la tabla con las diez mejores posiciones de acuerdo a la velocidad.
3. Finalmente, la vista de **perfil** la cual se mostrará toda la información del historial de jugadas del usuario.
4. **Perspectiva del producto**

IType se origina como resultado del trabajo a presentar para la materia “Ingeniería de Software” de la carrera Ingeniería en Computación de la Universidad Nacional de Córdoba.

Fue pensada como una aplicación de pasatiempos cuyo objetivo es obtener la mayor cantidad de palabras correctas ingresadas mediante el teclado de un teléfono del sistema Android.

La aplicación es de tipo código libre (open source) bajo la licencia “GNU General Public License versión 3.0”

Características:

* Soporte de diversas versiones de Android.
* Cuenta de usuario: cada instancia de aplicación está asociada con una cuenta en el sistema para su funcionalidad.
* Número de usuarios: el sistema da soporte a una gran cantidad de usuarios.
* Publicación de la prueba: La información relacionada a la prueba se publicará en la base de datos si el usuario luego de terminarla califica dentro de los primeros 10 puestos.

1. **Funciones del producto**

La función principal de la aplicación es la generación aleatoria de palabras en idioma español. Éstas se generarán de acuerdo al nivel que el usuario haya seleccionado en la vista de menú principal.

El nivel de dificultad tendrá incidencia sobre 5 factores en el mecanismo de generación de palabras:

* Frecuencia de palabras más o menos habituales del idioma.
* Frecuencia de letras más o menos utilizadas del idioma.
* Repetición de letras, en una palabra.
* Repetición de palabras.
* Tamaño de la palabra

La aplicación mostrará una palabra a la vez. Ésta se actualizará a medida que el usuario las escribe individualmente.

La duración de cada prueba es de 30 segundos. Iniciando la cuenta desde el instante en que se elige la dificultad y finalizando al cumplir los 30 segundos.

Una vez terminada la prueba, el sistema calculará la velocidad de “tipeo” que se obtuvo mediante la cantidad total de caracteres escritos correctamente y comparará con la base de datos si es apto para ser calificado en un puesto determinado.

Si es así, la aplicación determinará qué posición de tabla de puntajes es el correcto y se guardará la información en la base de datos MySql para luego poder ser mostrada por la aplicación en la vista de estadísticas.

1. **Clase y características del usuario**

El usuario de la aplicación es aquel que posee conocimientos básicos sobre la operación de dispositivos móviles que cuentan con el sistema Android.

La edad del usuario se sitúa como mayor a 5 años.

Debe poseer habilidad lectora.

Debe tener un buen conocimiento sobre el idioma español, ya que la aplicación estará exclusivamente en este idioma.

No es necesario ningún nivel de educación avanzada para usar esta aplicación ya que la interacción está en escribir correctamente (copiar caracteres).

Los usuarios deben utilizar el teclado con distribución QWERTY en su disposición hispanohablante (uso del carácter Ñ).

1. **Entorno operativo**

La presente aplicación está basada en el sistema Android. Android es un entorno operativo de código abierto destinado a dispositivos móviles. En este caso, la aplicación se diseña para la versión “smartphone”, sólo para teléfonos.

Android corre sobre un kernel de sistema operativo Linux.

Las versiones de Android actualmente soportadas de manera oficial son:

Android 7: “Nougat”

Android 8: “Oreo”

Android 9: “Pie”

Android 10: “Android Q” (en desarrollo)

Los requerimientos mínimos especificados por Android[[2]](#footnote-2) para el hardware son:

* RAM: 2 [GB]
* Almacenamiento interno: 16 [GB]
* Velocidad del procesador: 1.1 [GHz]
* Arquitectura de procesador: 32 bits.
* Protección con certificación IP: IP64
* Pantalla táctil

1. **Límites de diseño e implementación**

Este es un proyecto de código libre desarrollado exclusivamente para el sistema operativo Android.

* La aplicación debe ser compatible con la versión más reciente hasta la más antigua versión soportada por Android. (Hasta el momento: Android 7 a Android 9).
* Se deben utilizar herramientas de código abierto.
* Se deben seguir los lineamientos Material Design de estilo dados por el fabricante del sistema Android para cuestiones de visualización y estética de la aplicación en conjunción con la estética del sistema global.
* Uso de lenguaje Java para el desarrollo del código.
* Uso de base de datos mediante MariaDB.
* La base de datos contendrá cuatro tablas: una para la sesión de usuarios (usuario, contraseña) y las otras tres para los puestos con sus dificultades respectivas (usuario, velocidad).
* Uso de un sistema de manejo de datos para el control de la base de datos.
* El proceso de desarrollo se gestionará por medio de una determinada herramienta web para gestionar el proceso de desarrollo de software.
* Las pruebas de software se deberán gestionar con una [herramienta de gestión de software para testing](http://www.pmoinformatica.com/2015/04/herramientas-gestion-calidad-software.html).
* Uso de sistema de manejo de datos apropiado para el control de la base de datos.

1. **Documentación de usuario**

Se debe proveer una guía de uso (user guide – en inglés) de la aplicación en formato “.pdf” disponible al usuario.

Éste debe contener:

* Introducción de la aplicación
* Instalación
* Uso
* Reportar un problema.

1. **Dependencias y suposiciones**

La aplicación depende del correcto funcionamiento del sistema Android. Cualquier falla en dicho sistema puede afectar seriamente el funcionamiento de la aplicación

En primer lugar, se infiere que el usuario tiene conexión a Internet para poder descargar el archivo “.apk”.

Se espera que el espacio libre de almacenamiento sea el requerido para instalar la aplicación. Caso contrario, el sistema proveerá un mensaje de fallo de instalación.

También, se espera que el usuario haya activado la opción: “Permitir fuentes desconocidas” desde la aplicación Ajustes/Seguridad/Administración del dispositivo.

Esto permite instalar aplicaciones distintas a las existentes en el Google Play Store (tienda de aplicaciones).

Por último, es necesario que el teclado del sistema esté en distribución QWERTY – español.

**3. Requerimientos de interfaz externos**

1. **Interfaces de usuario**

* Uso del estilo Material Design que rige en el ecosistema Android.
* Orientación de pantalla: vertical.
* Botones:
* Ingresar (en vista de inicio de sesión)
* Registrarse (en vista de inicio de sesión y vista de registro)
* Salir (en vista de inicio de sesión)
* Volver (en vista de registro, vista de selección de nivel)
* Iniciar juego (en vista de menú principal)
* Perfil (en vista de menú principal)
* Salir (en vista de menú principal)
* Fácil (en vista selección de nivel)
* Medio (en vista selección de nivel)
* Difícil (en vista selección de nivel)
* Menú principal (en vista activa, vista de puestos y vista perfil)
* Comenzar (en vista activa)
* Reintentar (en vista activa)
* Siguiente (en vista de puestos)
* Fuente de texto: Roboto.

1. **Interfaces de hardware**

* Teléfonos con sistema Android.
* Compatibilidad: Android 7 o mayor.
* Uso de pantalla táctil y botones en dicha pantalla.
* Se delega el control de hardware al sistema Android.

1. **Interfaces de software**

* Java Development Kit 8.
* IDE: Android Studio 3.3.2
* Android 9.0 “Pie”.
* SQLite 3.28.0
* Gradle 4.1
* Librerías varias de Android para interfaz gráfica y funcionamiento.

1. **Interfaces de comunicación**

* Para la comunicación y administración entre servidor y la base de datos MySql se utilizará la herramienta phpMyAdmin.
* El servidor será implementado mediante el servicio de hosting de <https://ar.000webhost.com/>.
* El dominio de la página web será <http://itype.ml/> implementado con el servicio de dominios <https://www.freenom.com/es/index.html?lang=es>.

**4. Requerimientos funcionales**

La siguiente lista contiene todos los requerimientos funcionales que se deberán cumplir en la aplicación. Están identificados unívocamente con un nombre y una breve descripción.

1. **REQ 1: Entrada del usuario:**

**4.1.1 Descripción y prioridad**

Entrada de datos mediante teclado. Prioridad media. Se asigna dicha prioridad ya que el sistema Android limita directamente esta funcionalidad.

**4.1.2 Requerimientos funcionales**

**R1.1:** En la vista activa, la aplicación deberá permitir al usuario ingresar solo caracteres del tipo letra y espacios.  
  
**R1.2:** En la vista activa, la entrada será insensible a las mayúsculas por lo que se comparará siempre en minúscula.

**R1.3:** La palabra escrita será eliminada automáticamente al momento de ser reconocida.

**R1.4:** La entrada mediante teclado será únicamente caracter por caracter. La función de texto predictivo deberá estar deshabilitada dentro de la aplicación.

1. **REQ 2: Interfaz:**

**4.2.1 Descripción y prioridad**

Las vistas del programa especifican los distintos escenarios de funcionamiento, acceso y despliegue de datos.

Prioridad: Alta.

**4.2.2 Requerimientos funcionales**

**R2.1:** Las vistas deberán ser:

* Inicio sesión.
* Registro
* Menú principal
* Selección de nivel.
* Vista activa (Juego en marcha).
* Vista de puestos (estadísticas).
* Vista de perfil de usuario

**R2.2:** Tanto en la vista Inicio de sesión como en la vista de Menú Principal se deberá implementar un botón que permita al usuario salir de la aplicación.

**R2.3:** Tanto en la vista Activa como en la de Puestos y Perfil se deberá implementar un botón que permita al usuario ir a la vista de menú principal.

**R2.4:** En la vista de tabla de posiciones se deberán mostrar:

* El nombre del usuario
* El número del puesto en la tabla de posiciones
* La velocidad de la prueba en palabras por minuto más un plus por dificultad

**R2.5:** En la vista activa se deberá mostrar una sola palabra modelo a tipear.

**R2.6:** La tabla de puestos (estadísticas) deberá estar compuesta por los 10 puestos con mayor velocidad, ordenados de mayor a menor; siendo el primer puesto el más rápido.

**R2.7:** Se deberá permitir la elección de tres niveles de dificultad en la vista “selección de nivel”. Ellos son:

* Fácil.
* Intermedio.
* Difícil.

1. **REQ 3: Sesión:**

**4.3.1 Descripción y prioridad**

Se requiere garantizar el inicio de sesión de forma segura y confiable, así como la creación de la cuenta ante no estar registrado.

Prioridad: Alta.

**4.3.2 Requerimientos funcionales**

**R3.1:** Tanto la contraseña como el usuario se deberá codificar con el formato UTF-8.

**R3.2:** Se deberá poder iniciar y registrar en la base de datos.

**R3.3:** En caso de que el usuario exista y la contraseña no coincida con los datos almacenados; se debe indicar que ésta es incorrecta y se pedirá de nuevo un inicio de sesión.

1. **REQ 4: Funcionamiento general**

**4.4.1 Descripción y prioridad**

Características del juego en la vista activa en cuanto al puntaje, duración y forma de validación.

Prioridad: Alta

**4.4.2 Requerimientos funcionales**

**R4.1:** La prueba de velocidad deberá finalizar en un tiempo de 30 segundos,

**R4.2:** La entrada del usuario será reconocida como palabra para la comparación luego de haber escrito al menos un carácter seguido de un “enter”.

**R4.3:** La velocidad final se calculará como los caracteres totales “tipeados” por sesenta, dividido cinco veces el tiempo total de la prueba.

Esto es:



Siendo:

* **Vf**: Velocidad final. Medida en: palabras por minuto.
* **Tc**: Total de caracteres correctos “tipeados”. Medido en: caracteres.
* **Tp**: Tiempo total de la prueba. Medido en: segundos.

**R4.4:** Al finalizar el juego, se deberá subir los datos a la base de datos de forma automática.

**R4.5:** Al finalizar el juego, se deberá mostrar automáticamente la vista de

Puestos del juego en tiempo real.

**R4.6:** La actualización de Puestos (estadísticas) se implementará mediante un sistema de notificación el cual consiste en que el servidor deberá enviar la actualización de las tablas a todos los usuarios actualmente conectados y viendo las estadísticas.

**5. Casos de Uso**

1. **Definición de casos de uso**

* Usuario:

**CU1**: Salir

**CU2**: Registrarse

**CU3**: Iniciar sesión

**CU4**: Elección de la dificultad

**CU5**: Realizar la prueba

**CU6**: Realizar prueba de velocidad

**CU7**: Ver puestos

**CU8**: Ir a información de perfil

* Servidor:

**CS1**:Agregar usuario

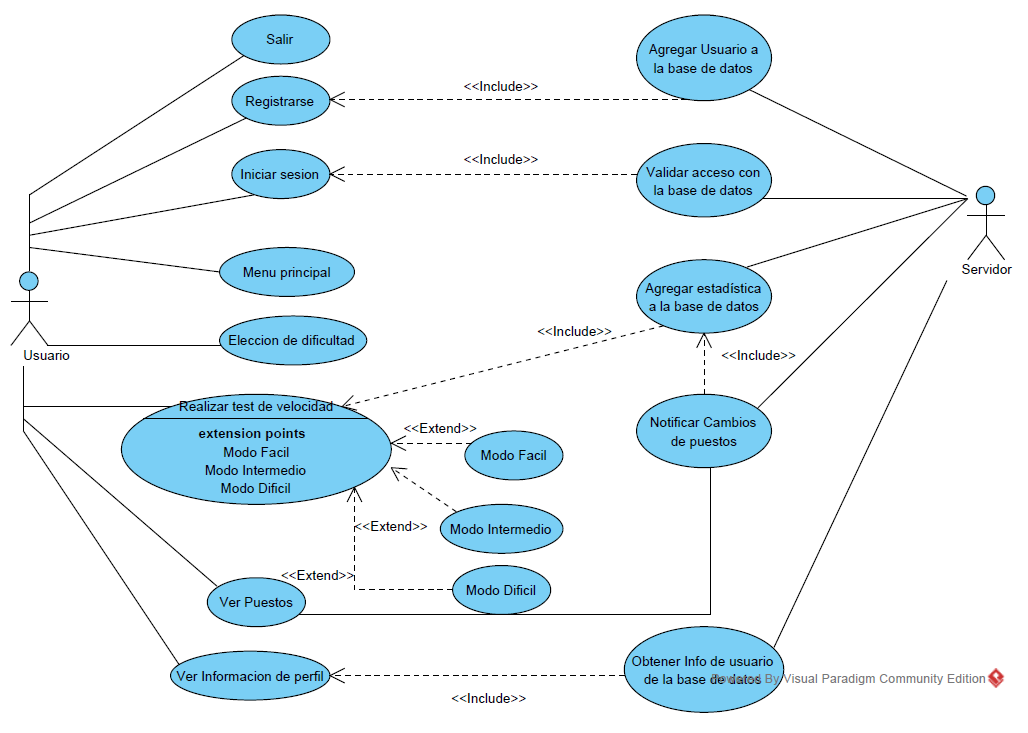
**CS2**:Validar acceso a la base de datos

**CS3**:Agregar datos de la prueba realizada en la base de datos

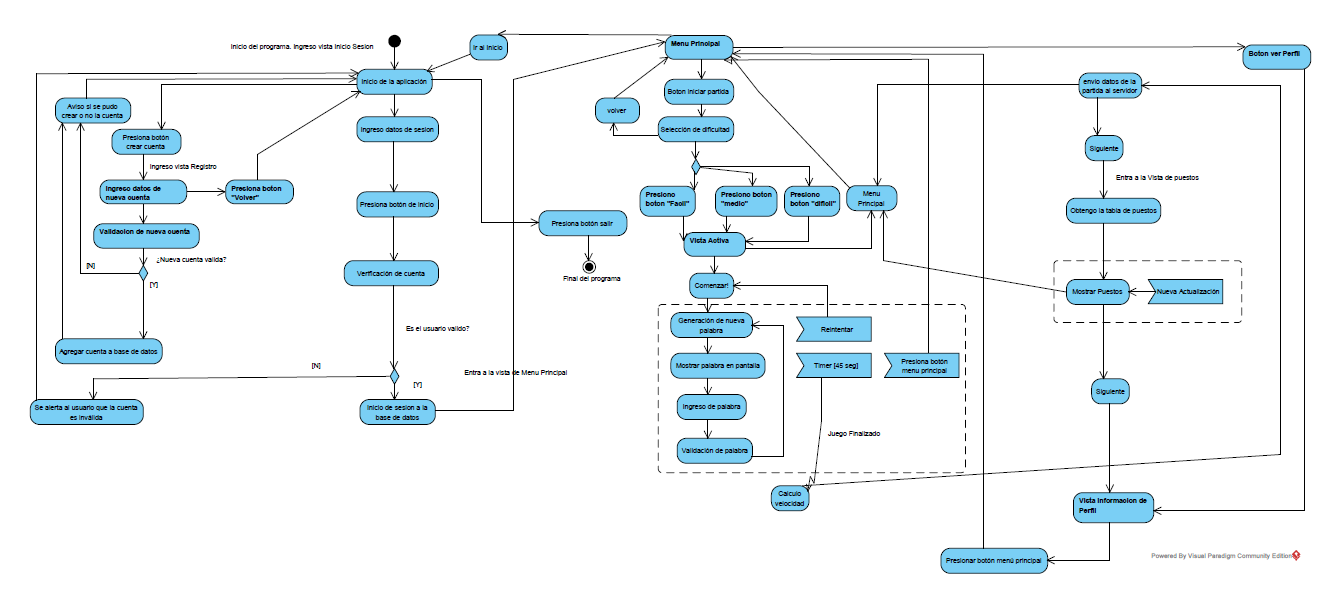
**CS4**:Notificar cambios de los puestos

**CS5**: Obtener información de usuario

1. **Diagrama de casos de uso**



1. **Diagrama de actividades**



1. **Matriz de trazabilidad**

Relación entre requerimientos funcionales y casos de uso

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Casos de Uso / Requisitos funcionales | CU1 | CU2 | CU3 | CU4 | CU5 | CU6 | CU7 | CU8 | CS1 | CS2 | CS3 | CS4 | CS5 |
| R1.1 |  |  |  |  |  | X |  |  |  |  |  |  |  |
| R1.2 |  |  |  |  |  | X |  |  |  |  |  |  |  |
| R1.3 |  |  |  |  |  | X |  |  |  |  |  |  |  |
| R1.4 |  |  |  |  |  | X |  |  |  |  |  |  |  |
| R2.1 |  |  | X |  | X | X | X | X |  |  |  |  | X |
| R2.2 | X |  |  | X | X | X |  |  |  |  |  |  |  |
| R2.3 |  |  |  | X |  | X | X | X |  |  |  |  |  |
| R2.4 |  |  |  |  |  |  | X | X |  |  |  |  |  |
| R2.5 |  |  |  |  |  | X |  |  |  |  |  |  |  |
| R2.6 |  |  |  |  |  |  | X | X |  |  |  |  |  |
| R2.7 |  |  |  |  | X |  |  |  |  |  |  |  |  |
| R3.1 |  | X | X |  |  |  |  |  | X |  |  |  |  |
| R3.2 |  | X | X |  |  |  |  |  | X | X |  |  |  |
| R3.3 |  | x | X |  |  |  |  |  | X |  |  |  |  |
| R4.1 |  |  |  |  |  | X |  |  |  |  |  |  |  |
| R4.2 |  |  |  |  |  | X |  |  |  |  |  |  |  |
| R4.3 |  |  |  |  |  | X |  |  |  |  |  |  |  |
| R4.4 |  |  |  |  |  | X | X |  |  | X | X |  |  |
| R4.5 |  |  |  |  |  | X | X |  |  |  |  |  |  |
| R4.6 |  |  |  |  |  |  | X |  |  |  |  | X |  |

**6. Requerimientos no funcionales**

1. **Requerimientos de rendimiento (performance)**

**RN1.1**: Toda funcionalidad del sistema debe responder al usuario en menos de 3 segundos.

**RN1.2:** El sistema debe tener una disponibilidad del 99% de las veces en que un usuario intente acceder

1. **Requerimientos de eficiencia**

**RN2.1**: La aplicación no deberá ocupar más de 200 Megabytes de espacio en disco

**RN2.2**: La aplicación deberá consumir más de 80 Megabytes de RAM.

**RN2.3**: La aplicación no deberá pedir accesos a datos del sistema alguno. Ejemplo: localización, contactos, acceso a mensajes, imágenes, etc.

**RN2.4**: La aplicación no deberá solicitar permisos de uso a cámara, micrófono u otros periféricos.

1. **Requerimientos de seguridad**

**RN3.1**: El sistema incluirá un procedimiento de autorización de usuarios, en el cual los usuarios deben identificarse usando un nombre de usuario y contraseña. Sólo los usuarios autorizados de esta forma podrán acceder a los datos del sistema.

1. **Atributos de calidad del software**

**RN4.1**: El promedio del tiempo de aprendizaje del sistema por un usuario deberá ser menor a quince minutos.

**7. Casos de prueba de sistema**

1. **Definición de casos de prueba de sistema**

**Identificador de caso de prueba:** TR1.1

**Descripción de la prueba:** verificar que no se pueden ingresar caracteres distintos a letra o espacio en la vista activa

**Funcionalidad a probar:** teclado en pantalla.

**Ejecución de la prueba (pasos):**

1. Abrir la vista activa.
2. Iniciar juego.
3. Situarse en la caja de texto
4. Observar el teclado y recorrer sus opciones durante el juego.

**Resultado esperado:** no presenta caracteres distintos a letras o espacios.

**Identificador de caso de prueba**: TR1.2

**Descripción de la prueba**: Verificar que letras en mayúscula o minúscula son indistintas.

**Funcionalidad a probar**: Algoritmo de validación de palabra.

**Ejecución de la prueba (pasos)**:

1. Abrir la vista activa.
2. Ingresar la palabra correcta intercalando letras en mayúscula y minúscula.
3. Presionar enter

**Resultado esperado**: la palabra ha sido aceptada correctamente por el sistema de validación de palabras.

**Identificador de caso de prueba**: TR1.3

**Descripción de la prueba**: palabra ingresada ha sido borrada luego de ingresar un espacio final.

**Funcionalidad a probar**: base de datos y algoritmo de validación de palabras

**Ejecución de la prueba (pasos)**:

1. Abrir vista activa.
2. Ingresar la palabra y un espacio correctamente.
3. Ingresar la siguiente palabra y enter correctamente.

**Resultado esperado**: verificación de ambas palabras correctamente.

**Identificador de caso de prueba**: TR1.4

**Descripción de la prueba**: verificar ingreso de texto de modo caracter por caracter.

**Funcionalidad a probar**: texto predictivo desactivado

**Ejecución de la prueba (pasos)**:

1. Abrir la vista activa.
2. Hacer click en comenzar.
3. Situarse en cuadro de entrada de texto.
4. Ingresar un caracter cualquiera.

**Resultado esperado**: el teclado no muestra palabras auto-sugeridas.

**Identificador de caso de prueba**: TR2.1

**Descripción de la prueba**: observar y enumerar las vistas requeridas.

**Funcionalidad a probar**: correcta definición y acceso a las vistas

**Ejecución de la prueba (pasos)**:

1. Abrir la aplicación
2. Ingresar con un usuario válido.
3. Seleccionar nivel fácil
4. Realizar la jugada
5. Finalizar el juego e ingresar a la vista de estadísticas
6. Volver a menú principal.

**Resultado esperado**: correcto acceso en orden y transferencia de las vistas.

**Identificador de caso de prueba**: TR2.2

**Descripción de la prueba**: probar el botón “salir” en vistas “Inicio de sesión” y “Selección de nivel”.

**Funcionalidad a probar**: botón Salir.

**Ejecución de la prueba 1 (pasos)**:

1. Abrir la vista inicio de sesión.
2. Divisar botón Salir.
3. Hacer clic en el botón “Salir”

**Resultado esperado**: se ha cerrado la aplicación.

**Ejecución de la prueba 2 (pasos)**:

1. Abrir la vista selección de nivel.
2. Divisar botón Salir.
3. Hacer clic en el botón “Salir”.

**Resultado esperado**: Se muestra la vista de Inicio de sesión

**Identificador de caso de prueba**: TR2.3

**Descripción de la prueba**: Transferencia de las distintas vistas al menú principal

**Funcionalidad a probar**: botón “Menú Principal”

**Ejecución de la prueba 1 (pasos)**:

1. Abrir la vista activa.
2. Hacer click en comenzar.
3. Divisar botón menú principal.
4. En la vista “Activa” hacer clic en el botón “Menú principal” y aceptar la advertencia

**Resultado esperado**: probar botón menú principal en la vista de perfil y puestos (estadísticas).

**Ejecución de la prueba 2 (pasos)**:

1. Abrir la vista activa.
2. Divisar botón menú principal.
3. En la vista “Activa” hacer clic en el botón “Menú principal” y aceptar la advertencia.

**Resultado esperado**: probar botón menú principal en la vista de Perfil y Puestos (estadísticas)

**Ejecución de la prueba 3 (pasos):**

1. Abrir la vista perfil
2. Divisar botón menú principal.
3. Realizar click al botón menú principal.

**Resultado esperado:** el sistema muestra el menú principal.

**Ejecución de la prueba 4 (pasos):**

1. Abrir la vista de los puestos.
2. Divisar botón menú principal.
3. Realizar click al botón menú principal.

**Resultado esperado:** el sistema muestra el menú principal.

**Identificador de caso de prueba**: TR2.4

**Descripción de la prueba**: observar los resultados de la jugada

**Funcionalidad a probar**: Estructura de la tabla de puestos.

**Ejecución de la prueba (pasos)**:

1. Abrir la vista de estadísticas.
2. Divisar columnas, nombre de usuario, puesto y velocidad

**Resultado esperado**: Información de puestos son mostrados en la tabla de

estadísticas de forma correcta.

**Identificador de caso de prueba:** TR2.5

**Descripción de la prueba**: Verificar la cantidad de palabras modelo

**Funcionalidad a probar**: algoritmo de selección y carga de palabra.

**Ejecución de la prueba (pasos):**

1. Abrir la vista activa
2. Hacer click en comenzar.
3. Divisar palabra modelo.

**Resultado esperado**: se observa una única palabra modelo por vez.

**Identificador de caso de prueba: TR2.6**

**Descripción de la prueba:** observar la organización de la tabla de estadísticas.

**Funcionalidad a probar**: registro de jugadas, uso, adición y obtención de datos de

jugadas a la base de datos estadística.

**Ejecución de la prueba 1 (pasos):**

1. Abrir la aplicación
2. Ingresar de forma válida al sistema
3. Seleccionar el nivel “fácil”
4. Hacer click en comenzar.
5. Esperar que termine la prueba.
6. Verificar puntaje cero. Observar la correcta posición en la tabla.

**Resultado esperado**: Orden de usuarios de mayor a menor en la tabla de puestos.

**Identificador de caso de prueba**: TR2.7

**Descripción de la prueba**: observar la vista de selección de nivel.

**Funcionalidad a probar**: Vista “selección de nivel”.

**Ejecución de la prueba 1 (pasos)**:

1. Abrir la vista de selección de nivel.
2. Divisar botones de dificultad.

**Resultado esperado**: Observar los botones “Fácil”, “Intermedio” y “Difícil”.

**Identificador de caso de prueba: TR3.1**

**Descripción de la prueba:** uso de formato UTF-8 para las contraseñas y usuarios.

**Funcionalidad a probar:** cifrado UTF-8 a la hora de hacer el http post al servidor.

**Ejecución de la prueba (pasos):**

1. Abrir la aplicación
2. Abrir aplicación de sniffer de paquetes de red.
3. Registrar un nuevo usuario.

**Resultado esperado:** Se observa la transferencia de información de usuario encriptada

**Identificador de caso de prueba**: TR3.2

**Descripción de la prueba**: obtener datos, comparar y validar usuario y contraseña con información en la base de datos de usuario.

**Funcionalidad a probar**: algoritmo de validación de cuenta.

**Ejecución de la prueba (pasos)**:

1. Abrir la aplicación
2. Registrar usuario inexistente
3. Iniciar sesión con un usuario ya registrado
4. Verificar que se pueda ingresar correctamente

**Resultado esperado**: Inicio de sesión exitoso

**Ejecución de la prueba 2(pasos):**

1. Abrir la vista de registro
2. Registrar usuario existente

**Resultado esperado:** Se observa alerta de usuario ya registrado.

**Ejecución de la prueba 3(pasos):**

1. Abrir la aplicación
2. Iniciar con un usuario con caracteres no válidos.

**Resultado esperado:** Se observa alerta de campo vacío o nombre de usuario inválido.

**Ejecución de la prueba 4(pasos):**

1. Abrir la aplicación
2. Iniciar con un nombre de usuario o contraseñas en blanco.

**Resultado esperado:** Se observa alerta de campo vacío o nombre de usuario

inválido

**Identificador de caso de prueba**: TR3.3

**Descripción de la prueba**: Detectar y avisar a usuarios no registrados en la base de datos que la cuenta no es válida.

**Funcionalidad a probar**: algoritmo de validación de cuenta, base de datos, comparación con la base de datos.

**Ejecución de la prueba (pasos)**:

1. Iniciar aplicación.
2. Ingresar un usuario no registrado actualmente

**Resultado esperado**: Mensaje de “Usuario inválido en pantalla”

**Identificador de caso de prueba**: TR4.1

**Descripción de la prueba**: se verifica la duración temporal del juego.

**Funcionalidad a probar**: Temporizador de prueba

**Ejecución de la prueba (pasos)**:

1. Abrir la aplicación
2. Ingresar de forma válida al sistema
3. Seleccionar botón “Fácil”.
4. Verificar el tiempo hasta que la prueba finaliza, (se transfiere a la vista de estadística)

**Resultado esperado**: La duración de la transferencia de vistas es de 30 segundos.

**Identificador de caso de prueba**: TR4.2

**Descripción de la prueba**: se prueba la correcta transición de palabra a otra nueva palabra.

**Funcionalidad a probar**: algoritmo de validación de palabra

**Ejecución de la prueba (pasos)**:

1. Abrir la vista activa
2. Hacer click en comenzar.
3. Ingresar la palabra modelo correctamente
4. Presionar enter

**Resultado esperado**: Observar que la palabra modelo se actualizó.

**Identificador de caso de prueba**: TR4.3

**Descripción de la prueba**: se probará la correcta obtención del puntaje.

**Funcionalidad a probar**: algoritmo de validación, obtención del puntaje, asociación de datos.

**Ejecución de la prueba (pasos)**:

1. Abrir la vista activa.
2. Hacer click en comenzar.
3. Realizar la prueba contando los caracteres correctos ingresados.

**Resultado esperado**: Verificar que se cumpla la ecuación de puntaje

**Identificador de caso de prueba**: TR4.4

**Descripción de la prueba**: el fin de cada jugada implica subir datos a la base de datos de estadística en forma automática.

**Funcionalidad a probar**: algoritmo de obtención de datos estadísticos, base de datos de resultados

**Ejecución de la prueba (pasos):**

1. Abrir la aplicación
2. Ingresar de forma válida al sistema
3. Ir al perfil y observar la información de usuario.
4. Volver al menú principal.
5. Seleccionar botón “Fácil”.
6. Realizar jugada.
7. Clic siguiente en la vista de estadísticas para ver la información del usuario.

**Resultado esperado:** Se observa la actualización de la información de usuario.

**Identificador de caso de prueba**: TR4.5

**Descripción de la prueba**: prueba de vista resultante al finalizar el tiempo de jugada

**Funcionalidad a probar**: algoritmo de validación, cronómetro, vistas.

**Ejecución de la prueba (pasos)**:

1. Abrir la vista activa.
2. Clic en comenzar.
3. Esperar que la prueba finalice
4. Resultado esperado: despliegue de vi

**Resultado esperado**: despliegue de vista Puestos

**Casos de prueba para requerimientos no funcionales**

**Identificador de caso de prueba**: TRN1.1

**Descripción de la prueba**: evaluación de eficiencia temporal del sistema.

**Funcionalidad a probar**: tiempo de respuesta.

**Ejecución de la prueba (pasos)**:

1. Iniciar la herramienta de testing en tiempo real.
2. Iniciar la aplicación.
3. Realizar el paso por las vistas.
4. Observar el tiempo transcurrido entre cada acción.

**Resultado esperado**: cada acción no debe tomar más de 3 segundos.

**Identificador de caso de prueba**: TRN1.2

**Descripción de la prueba**: prueba de actualización de base de datos

**Funcionalidad a probar**: base de datos de usuario, palabras y estadísticas.

**Ejecución de la prueba (pasos)**:

1. Iniciar aplicación
2. Iniciar sesión correctamente
3. Elegir dificultad fácil
4. Esperar que la prueba finalice
5. Verificar el tiempo desde que la prueba finaliza hasta que se inicia la vista de estadísticas.

**Resultado esperado**: tiempo de actualización menor a 2 segundos.

**Identificador de caso de prueba**: TRN2.1-2

**Descripción de la prueba**: prueba de eficiencia del sistema

**Funcionalidad a probar**: uso de espacio en disco y memoria RAM.

**Ejecución de la prueba (pasos)**:

1. Inicio de Android Studio
2. Inicio de aplicación
3. Inicio del monitor de memoria
4. Correr el sistema y realizar un juego.

**Resultado esperado**: uso de espacio en disco menor a 200 MB y memoria RAM menor a 80 [MB]

**Identificador de caso de prueba**: TRN2.3-4

**Descripción de la prueba**: prueba de eficiencia y acceso a recursos.

**Funcionalidad a probar**: acceso a periféricos del teléfono o datos de sistema.

**Ejecución de la prueba (pasos)**:

1. Instalar la aplicación
2. Iniciar aplicación
3. Iniciar sesión
4. Realizar juego
5. Salir de la aplicación

**Resultado esperado**: no petición de acceso a elementos de sistema o periféricos

**Identificador de caso de prueba**: TRN4.1

**Descripción de la prueba**: accesibilidad.

**Funcionalidad a probar**: facilidad e intuitividad del sistema.

**Ejecución de la prueba (pasos)**:

1. Iniciar cronómetro
2. Iniciar aplicación
3. Entregar a usuario sin previo contacto con la aplicación
4. Observar la interacción del mismo
5. Esperar a que el usuario pueda realizar al menos un test correctamente

**Resultado esperado**: tiempo transcurrido para la comprensión de la aplicación menor a 15 minutos.

1. **Definición de casos de prueba alternativos**

**Identificador de caso de prueba**: TA1.1

**Descripción de la prueba**: Prueba de velocidad extrema.

**Funcionalidad a probar**: velocidad del algoritmo de generación de palabras y respuesta de la aplicación ante una situación atípica.

**Ejecución de la prueba (pasos)**:

1. Iniciar aplicación
2. Iniciar sesión
3. Realizar la prueba con una velocidad incremental de caracteres correctos por minuto.

**Resultado esperado**: Verificar el límite de la respuesta de la aplicación ante velocidades superiores a la capaz por cualquier ser humano.

**Identificador de caso de prueba**: TA1.2

**Descripción de la prueba**: Forzar caracteres no permitidos por la aplicación

**Funcionalidad a probar**: algoritmo de validación de palabras

**Ejecución de la prueba (pasos)**:

1. Iniciar aplicación
2. Iniciar sesión
3. Realizar la prueba forzando caracteres inusuales.

**Resultado esperado**: Correcto funcionamiento del sistema de validación.

**Identificador de caso de prueba**: TA1.3

**Descripción de la prueba**: Realizar la prueba e ir a la vista de estadísticas con una cantidad paralela de usuarios incremental.

**Funcionalidad a probar**: Sistema de actualización de estadísticas por parte del servidor.

**Ejecución de la prueba (pasos):**

1. Iniciar vista de perfil.
2. Verificar la información mostrada.
3. Generar nuevo thread.
4. Comenzar el thread con tarea abrir nueva vista de perfil.

**Resultado esperado:** Verificar la capacidad de manejo de los http request

concurrentes con el servidor web.

1. **Pruebas de aceptación**

De los casos de prueba establecidos en el apartado anterior, se eligen como pruebas de aceptación para el usuario las pruebas:

|  |  |  |  |
| --- | --- | --- | --- |
| * TR1.1 | * TR2.4 | * TR4.1 | * TRN2.1-2 |
| * TR1.2 | * TR2.6 | * TR4.4 | * TRN4.1 |
| * TR1.3 | * TR3.1 | * TR4.5 | * TRA1.3 |
| * TR1.4 | * TR3.2 | * TRN1.1 |
| * TR2.1 | * TR3.3 | * TRN1.4 |

1. **Pruebas smoke**

|  |
| --- |
| * TR2.1 |
| * TR2.4 |
| * TR2.6 |
| * TR3.2 |
| * TR3.3 |
| * TR4.4 |
| * TRN1.3 |

1. **Matriz de trazabilidad**

Relación entre requerimientos funcionales y casos de prueba

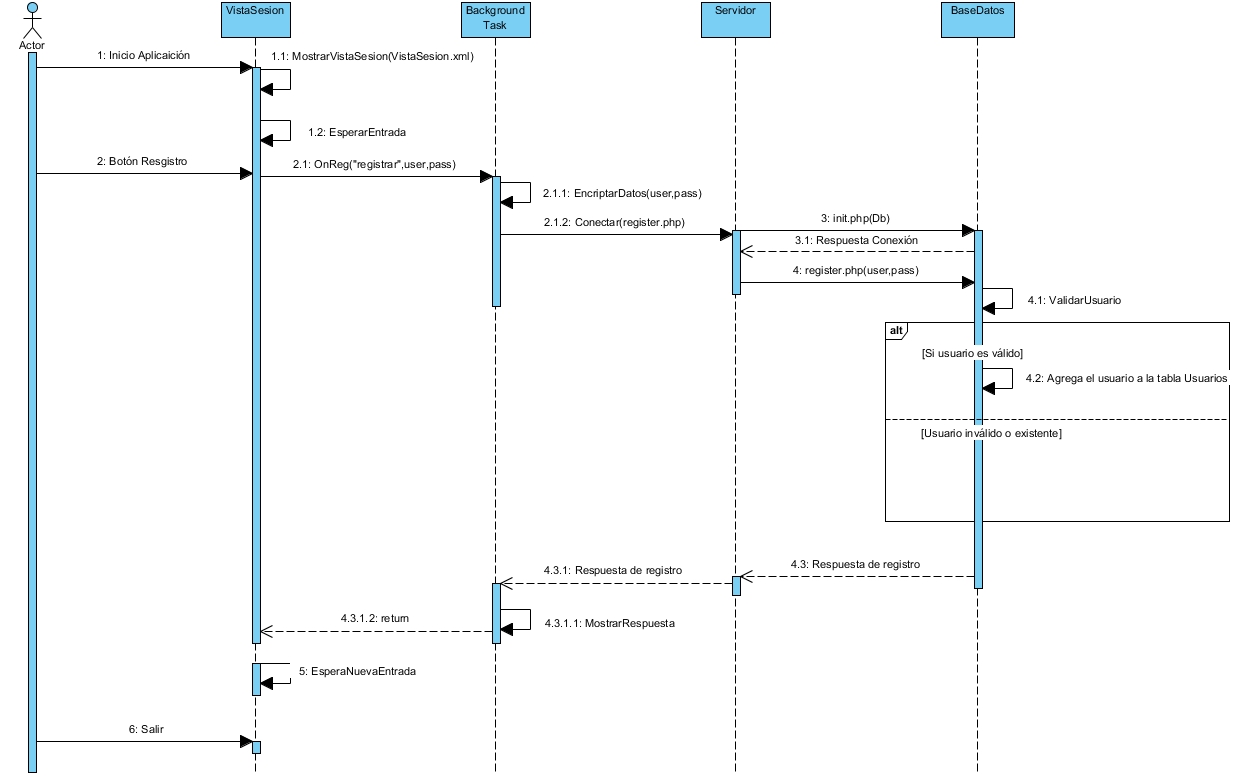
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Casos de prueba / Requermientos | TR1.1 | TR1.2 | TR1.3 | TR1.4 | TR2.1 | TR2.2 | TR2.3 | TR2.4 | TR2.5 | TR2.6 | TR2.7 | TR3.1 | TR3.2 | TR3.3 | TR4.1 | TR4.2 | TR4.3 | TR4.4 | TR4.5 | TRN1.1 | TRN1.2 | TRN1.3 | TRN2.1-2 | TRN2.3-4 | TRN4.1 | TA1.1 | TA1.2 | TA1.3 |
| R1.1 | X |  |  |  |  |  |  |  | X | X |  |  |  |  |  | X | X | X |  |  |  |  |  |  |  |  |  |  |
| R1.2 |  | X |  |  |  |  |  |  | X |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |
| R1.3 |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |
| R1.4 | X | X | X | X |  |  |  |  | X |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |
| R2.1 |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |
| R2.2 |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.3 |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.4 |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.5 |  |  | X |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.6 |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R2.7 | X | X | X |  | X |  | X | X | X | X | X |  |  |  | X |  | X | X | X |  |  |  |  |  |  | X | X | X |
| R3.1 | X |  |  | X |  |  |  |  |  |  |  | X | X |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |
| R3.2 | X | X | X | X | X | X | X | X | X | X | X | X | X |  | X | X | X | X | X |  |  |  |  |  |  | X | X | X |
| R3.3 |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R4.1 |  |  |  |  |  |  |  | X |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  | X |
| R4.2 | X | X | X |  |  |  |  |  | X | X |  |  |  |  |  | X | X | X | X |  |  |  |  |  |  |  |  |  |
| R4.3 |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |
| R4.4 |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |
| R4.5 |  |  |  |  | X |  |  | X |  | X |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |
| R4.6 |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |

Relación entre requerimientos no funcionales y casos de prueba

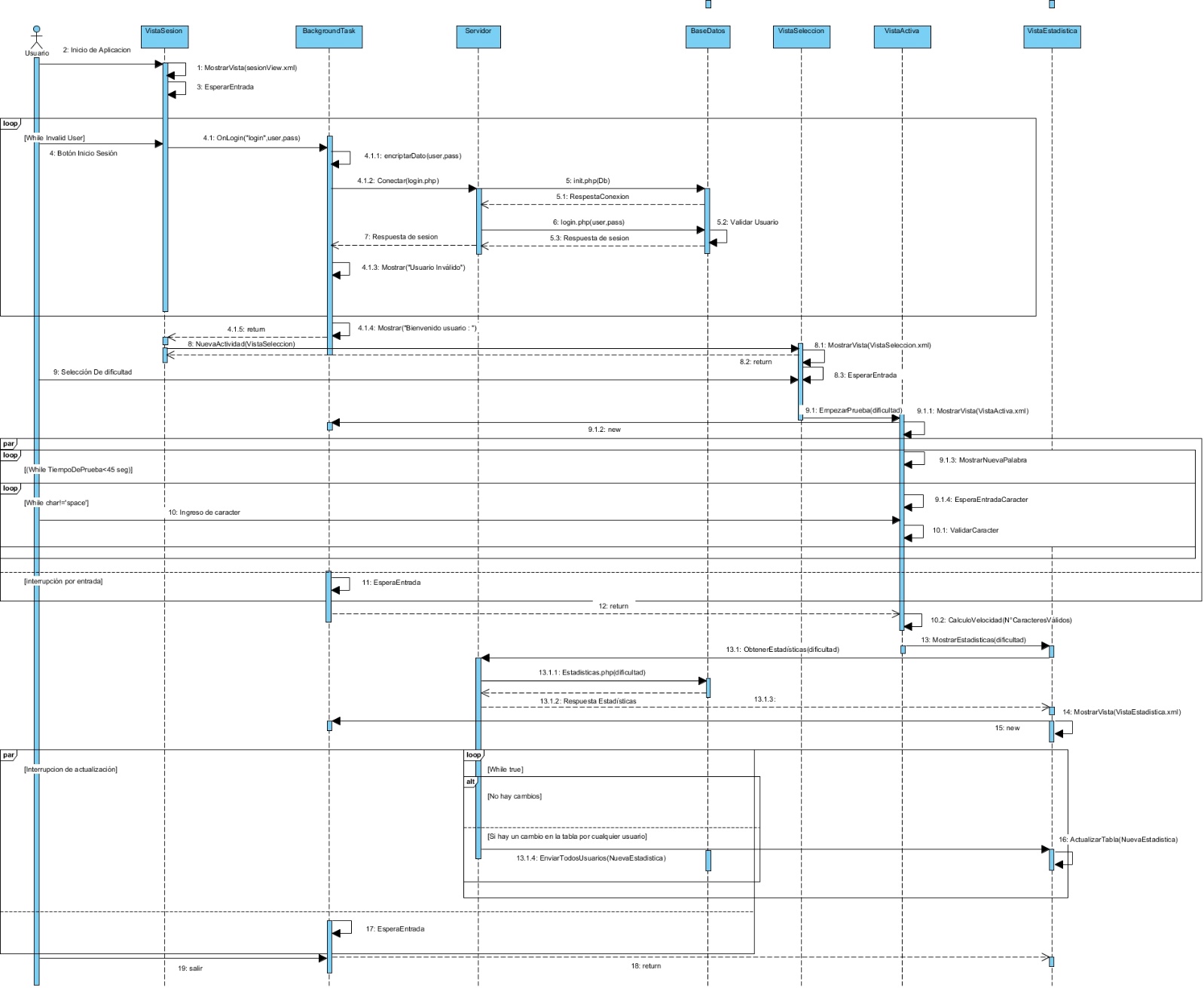
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Casos de prueba / Requermientos | TR1.1 | TR1.2 | TR1.3 | TR1.4 | TR2.1 | TR2.2 | TR2.3 | TR2.4 | TR2.5 | TR2.6 | TR2.7 | TR3.1 | TR3.2 | TR3.3 | TR4.1 | TR4.2 | TR4.3 | TR4.4 | TR4.5 | TRN1.1 | TRN1.2 | TRN1.3 | TRN2.1-2 | TRN2.3-4 | TRN4.1 | TA1.1 | TA1.2 | TA1.3 |
| RN1.1 | X | X | X | X | X | X | X |  | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |  | X |  | X |
| RN1.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  | X |  |
| RN1.3 |  |  |  |  | X | X | X |  | X | X | X | X | X | X | X | X | X | X | X | X | X | X |  |  |  |  | X | X |
| RN2.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |
| RN2.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |
| RN2.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |
| RN2.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |
| RN3.1 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |  |  | X | X | X | X |
| RN4.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |

**8. Diagramas adicionales**

1. **Diagrama de secuencia para el registro de usuario**

****

1. **Diagrama de secuencia para el juego**

****

**9. Arquitectura preliminar**

Se presenta un diagrama que representa un primer bosquejo de la idea de cómo será la arquitectura de nuestro sistema.

Se denota que utilizaremos el patrón de arquitectura MVC (Modelo – Vista – Controlador).

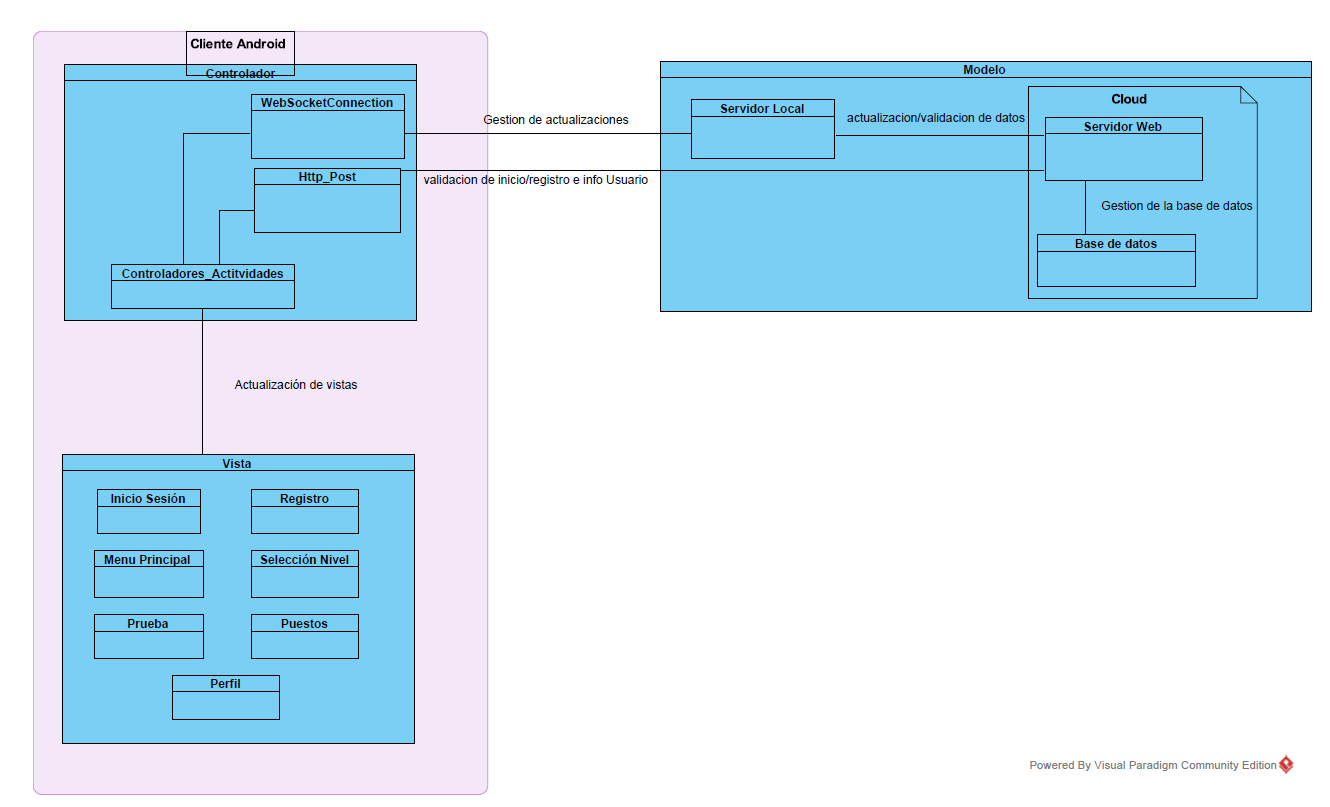
Esto se justifica ya que es el mejor patrón que se ajusta al tipo de aplicación que se

desarrollará. Teniendo una parte de presentación de datos (vista), otra que recibe

interacción de usuario (controlador) y finalmente, el contenedor y cálculo de los datos (modelo).

Las vistas permiten cumplir los requerimientos de presentación de palabras, credenciales y menús. Modelos aportan el almacén y procesamiento de datos tales como credenciales, procesamiento de palabras en base a la dificultad, cálculo de posiciones, velocidad instantánea, etc. Controladores soportan la interacción con el usuario, por ejemplo, acciones por botones o cajas de texto de entrada.

**Diagramas de arquitectura preliminar**



1. Instituto de Ingeniería Eléctrica y Electrónica - Institute of Electrical and Electronics Engineers, en inglés. Plantilla de Documento de Requerimientos. <https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc> [↑](#footnote-ref-1)
2. Para Android 7 “Nougat”. Android. https://www.android.com/enterprise/recommended/requirements/ [↑](#footnote-ref-2)