

Sistema de adquisición y procesamiento de datos de un guante inteligente

Sergio Giménez Osuna

Resumen— En la actualidad todo el mundo dispone de tecnología móvil la cual es capaz de realizar una gran cantidad de tareas, desde entretenimiento hasta finanzas. En el sector de la salud, los móviles son el medio ideal para el seguimiento del estado de un paciente; desde monitorización de medicación pasando por la toma de valores, su portabilidad permite una gran flexibilidad a los usuarios. En este proyecto se propone realizar un sistema de adquisición de datos la cual permita a un especialista monitorizar los ejercicios de un paciente el cual usa un guante inteligente, de manera que el especialista pueda hacer un seguimiento y corregir al paciente en corto periodo de tiempo.

Palabras clave— Bluetooth, aplicación android, FRDM-KL25Z, monitorización de pacientes, web, guante inteligente.

Abstract— Nowadays everyone disposes of mobile technology which is able to perform a great amount of tasks, from entertainment to finances. In the health sector, mobile phones are ideal to trace the state of a patient; from medication monitorization to information collector its portability brings great flexibility to the users. In this project, it is intended to create a complete data acquisition system which allows a specialist to trace the exercises of a patient that uses a smart glove, in such a way that the specialist is able to correct the patient in a short span of time.

Index Terms— Bluetooth, android application, FRDM-KL25Z, patient monitorización, web, smart glove



1 INTRODUCCIÓN

La tecnología se ha vuelto una parte casi imprescindible de una gran parte del mundo actual, esta nos permite realizar un gran número de actividades y tareas: entretenimiento, información, comunicación, etc.

Entre estas actividades se encuentra las de la salud, como por ejemplo, la monitorización de las constantes vitales de un paciente o el diagnóstico de posibles enfermedades que una persona puede tener.

En nuestro caso, la tecnología nos permite realizar una tarea como la monitorización de pacientes de manera remota; mientras que es algo aparentemente simple, esta mejora la calidad de vida de los usuarios gracias a la comodidad que aporta.

Uno de los problemas encontrados usualmente en rehabilitación es que los pacientes no se adhieren a los planes del terapeuta y éste no tiene información sobre el progreso del paciente, por lo tanto se ha diseñado un

sistema basado en la monitorización de los ejercicios que el terapeuta prescribe que a la vez permite al usuario tener información sobre el progreso de estos ejercicios.

Para ello contamos de un guante inteligente realizado en proyectos previos por Roger Recasens realizo una mejora sobre el guante inteligente [1] realizado por Jordi González [2].

El sistema de proyecto consta de los componentes presentados en la figura 1. Los anteriores proyectos estuvieron centrados en el guante mientras que este se ha centrado mas en la aplicación. Con tal de facilitar la usabilidad del sistema se hace uso de conexiones inalámbricas para comunicar todos los componentes. También se dispondrá de una página web con la que el fisioterapeuta pueda ver incidencias en un corto periodo de tiempo para poder corregirlas.

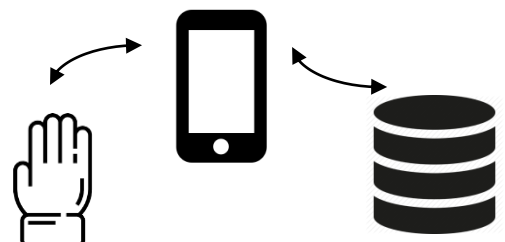


Figura 1: Sistema del proyecto

- E-mail de contacto: sergiogios@hotmail.com
- Mención realizada: Ingeniería de Computadores
- Trabajo tutorizado por: Dolores Rexachs (Departamento de Arquitectura de Computadores y Sistemas Operativos)
- Curso 2018/2019

2 OBJETIVOS

Como objetivo general tendremos diseñar e implementar un sistema para monitorizar la actividad del guante inteligente usado en la rehabilitación de pacientes.

A continuación se resumen los objetivos específicos a realizar en este proyecto:

- **Crear una aplicación móvil para el usuario y una web para el especialista**

Realizar una interfaz la cual permita al usuario ver en tiempo real cual es el estado de la sesión, mostrando el total de las fuerzas realizadas a cada paso mediante un grafo.

Con tal que el especialista pueda revisar las sesiones de cada paciente, se creará una página web en la cual se podrá consultar todas las sesiones de todos los usuarios con la correspondiente información asociada a estas.

- **Comunicar móvil, guante y servidor**

Crear los módulos de comunicación guante con móvil y móvil con servidor. Mientras que la comunicación guante-móvil se deberá realizar por Bluetooth, la comunicación móvil-servidor será por datos o WiFi.

- **Mantener datos de una sesión**

Debido a la posibilidad que un usuario no pueda subir su sesión por falta de conexión, se deberá crear un módulo que permita guardar una sesión en la memoria del móvil.

- **Crear una base de datos**

Se realizará una base de datos en SQL con el cual poder mantener las múltiples sesiones de los usuarios.

3 ESTADO DEL ARTE

En la actualidad existen múltiples tipos de guantes de rehabilitación, mientras algunos son mecánicos otros incorporan elementos electrónicos como el nuestro.

Respecto los mecánicos [3], mientras que no sufren de ningún desgaste electrónico y su vida útil puede ser algo más larga, estos no proporcionan ningún tipo de feedback al especialista y por lo tanto, su utilidad en terapias de rehabilitación se ve algo mermada.

En cuanto a los que incorporan elementos electrónicos [4], como el usado en este proyecto, existe cierta variedad de

modelos en el mercado comercial; sin embargo estos están más dedicados al ámbito de la auto-rehabilitación mediante el uso de aplicaciones lúdicas con aplicaciones moderadas de supervisión. Por otra parte usualmente, estos guantes tienen un alto precio de mercado, hecho el cual dificulta la compra por parte de particulares y especialistas de la salud.

4 METODOLOGÍA

En este proyecto se ha utilizado el modelo de metodología incremental. De esta manera se han podido implementar cada uno de los módulos del sistema, comprobando que funcionan de manera correcta y sin imprevistos.

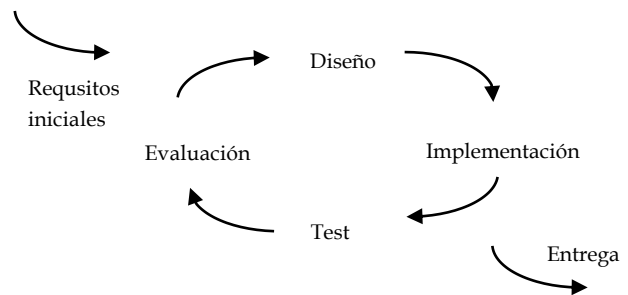


Figura 2: Metodología del proyecto

Requisitos iniciales: Se recopilan tanto los objetivos como los requisitos a cumplir en el proyecto. Se definen los módulos que deberán ser creados y como éstos se relacionan entre ellos. Se intenta prever todos los posibles contratiempos asociados a los módulos.

Requerimientos: Se analizan todos los objetivos que se han previsto para este módulo y se estudia como se llevará a cabo la implementación del módulo.

Implementación: En esta fase se procederá a implementar las funcionalidades del módulo según las especificaciones previstas.

Test: Una vez implementado el módulo se procede a realizar pruebas sobre éste, probando los posibles casos que el usuario pueda realizar con tal de evitar posibles problemas una vez este comience a usar la aplicación.

Evaluación: Finalizada la implementación y test del módulo se revisa los requerimientos con tal de asegurar que podemos pasar a la siguiente iteración del próximo módulo.

Entrega: Una vez finalizado el proyecto podemos realizar la entrega al personal correspondiente.

Debido que a veces era necesario un módulo para realizar pruebas con otro, siempre se optó por escoger datos aproximados con los que realizar los test con tal de no modificar el flujo de la metodología.

5 DISEÑO DEL SISTEMA

La estructura del sistema se basa en tres elementos principales:

- El guante: Se encarga de recoger, procesar y enviar al móvil
- La aplicación móvil (App): Se encarga de recopilar los datos del guante, presentarlos, enviarlos y gestionar los usuarios
- La aplicación web: Se encarga almacenar(Base de datos) y presentar los datos de los usuarios.

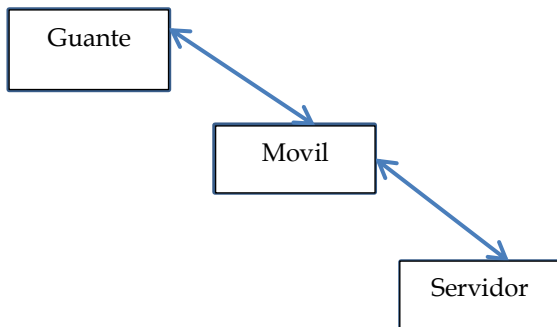


Figura 3: Comunicación entre fuerzas

5.1 Guante

El guante es un dispositivo formado por tres elementos principales:

- Placa FRDM-KL25Z: De la empresa Freescale, contiene un procesador Cortex-M0 este se encargara de controlar tanto el módulo bluetooth como los sensores de fuerza.
- Módulo Bluetooth RN-41: Con una posibilidad de envío de 3Mbps y diferentes modos de consumo, será el encargado del envío de datos a móviles.



Figura 4: Módulo Bluetooth RN-41

- Sensores FlexiForce A201: Contaremos con 5 tiras cada una para un dedo de la mano y realizarán la tarea de recoger las fuerzas realizadas de cada dedo.



Figura 5: Sensor FlexiForce

5.1.1 Lógica del guante

Podemos desglosar las actividades que realiza el guante en dos apartados: Por una parte la recogida de datos y por otra el procesado con el envío correspondiente de estos.

En la parte de recogida de datos tendremos funciones que escanean la fuerza realizada a cada sensor; al ser un sensor resistivo, deberemos transformar la resistencia en gramos para que pueda ser inteligible.

Una vez finalizada la recopilación en un instante de tiempo se procederá a transformar las fuerzas en una única cadena de carecteres.

P50.1I32.2C41.1A54.7M76.1F#

Figura 6: Cadena ejemplo de fuerzas

Esta será la cadena base con la que trabajaremos en todas los elementos que componen el proyecto. En ella se indica la fuerza asociada a un dedo [**P** = 50.1, **I** = 32.2], el final de la cadena indicado por el símbolo [#]. Ademas se incopora el carácter [**F**] utilizado para poder fragmentar la cadena en cada una de las fuerzas asociadas a cada dedo.

De esta manera, a lo largo del proyecto siempre que se requiera el acceso de una fuerza se realizara mediante la resta de punteros asociados a las letras.

5.1.2 Comunicación con el movil

El guante al principio se preparara, discerniendo el estado en el que está con el LED que la placa trae incorporada. Mientras que naranja indica que la placa aún se está preparando, con tal de evitar el malfuncionamiento de el módulo Bluetooth, cuando éste se apaga hay una indicación de que ya se puede iniciar las comunicaciones.

Debido a que el inicio de una sesión se puede realizar de manera asíncrona, el guante esperará hasta que reciba una señal del móvil, dicha señal indica que el móvil está preparado para recibir un dato, en nuestro caso, la cadena de la figura 6. Una vez enviada la cadena en el bufer, el guante volverá al bucle inicial esperando la señal del móvil.

5.2 Aplicación móvil

La aplicación ha sido creada para dispositivos móviles con sistema operativo Android. De cara al usuario, la aplicación tiene capacidad para mostrar las fuerzas que se realizan en una sesión con el guante, iniciar sesión con un usuario propio y subir o eliminar una sesión previa guardada.

5.2.1 Comunicación con el guante

Con tal de realizar las comunicaciones se ha creado un Singleton para facilitar el acceso de diversas actividades al mismo canal de comunicación. Para establecer el canal de comunicación, se crea un socket y se conecta con el módulo Bluetooth del guante. Una vez realizada la conexión se establece el canal. Debido a que necesitaremos manejar tanto la UI (interfaz) como los datos, estos deberán ser tratados asíncronamente mediante un Handler de Android.

Cuando un usuario presiona el botón de actualización para ver la fuerza que está realizando se inicia un amplio proceso de tareas. Para empezar, se envía por el canal un "ping" al guante para que recoja datos y los envíe de vuelta. Una vez el Handler que espera a que llegue un dato, detecta este en el buffer, inicia una alerta con el EventBus para enviar el dato a la actividad principal y finalmente elimina la cadena procesada del buffer.

5.2.2 EventBus

EventBus [5] es una librería libre la cual permite facilitar las comunicaciones entre varias actividades, en un estilo productor/subscriptor. Se optó por esta librería debido a que en el momento de la búsqueda era la única que encajaba con la funcionalidad deseada.

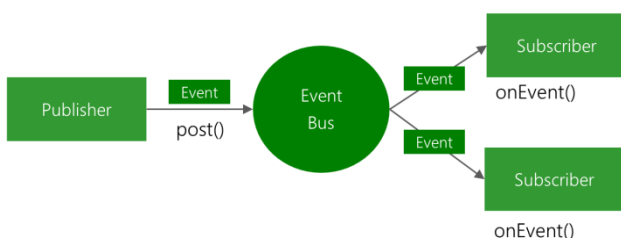


Figura 7: Esquema de EventBus 1

De esta manera, nuestra actividad que mantiene la canal sería el productor y la actividad principal/interfaz sería el

subscriber. Una vez el dato es enviado por el EventBus, la actividad principal detectará el evento y iniciará un conjunto de tareas para actualizar la pantalla y mantener la cadena de la sesión.

5.2.3 Actualizando la interficie

Una vez llega la señal de datos a través del event bus iniciamos la parte final de la actualización la UI.

Primero, cada vez que una cadena llegue la añadiremos a las otras que haya en la sesión actual, de esta manera mantenemos el estado de una sesión como el conjunto de cadenas recibidas. Una vez realizado la adición procedemos a procesar la cadena en cada una de las fuerzas de los distintos dedos, estas son guardadas en una clase para su posible posterior recogida o uso. Parseados los datos, asignamos los nuevos valores al grafo y lo actualizamos.

5.2.4 Muestra de las fuerzas

Para mostrar el grafo se ha utilizado la librería libre MPAndroidChart [6] la cual permite utilizar un amplio número de grafos y personalización. Se optó por esta librería debido a que a primera vista tenía una gran variedad de grafos y fácil implementación.

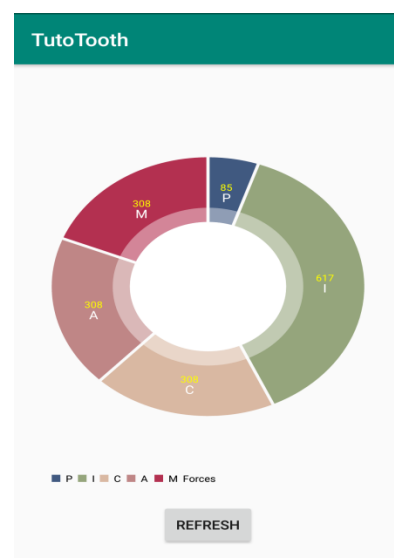


Figura 8: Ejemplo de grafo utilizado

Con este grafo, cada dos pulsaciones del botón de refresco se mostrara las fuerzas de la última pulsación. Se ha optado por el uso de un pie chart con leyenda para facilitar el reconocimiento de las fuerzas por el usuario en cada momento.

También, para evitar confusión para el usuario, antes de

iniciar la sesión, se muestra un grafo con valores preasignados.

Finalmente, para eliminar falsas lecturas de fuerzas no se muestra ninguna fuerza que no supere los diez gramos.

5.2.5 Finalizando la sesión

Cuando el usuario haya realizado todas las repeticiones necesarias, este lo hará notar pulsando el botón de finalización de sesión.

En el caso que el usuario este logeado con usuario y disponga de datos móviles o WiFi, la cadena de la sesión se guardara automáticamente en la base de datos del servidor.

Debido a que en un determinado momento es posible que un usuario no disponga de conexión WiFi o datos mobiles se ha diseñado un sistema mediante SharedPreferences para mantener la cadena de la última sesión. Esto conlleva ciertas limitaciones para el usuario, como por ejemplo, restringir el inicio de otra sesión si no se ha subido previamente la ultima, debido a que solo se almacena una sesión a la vez.

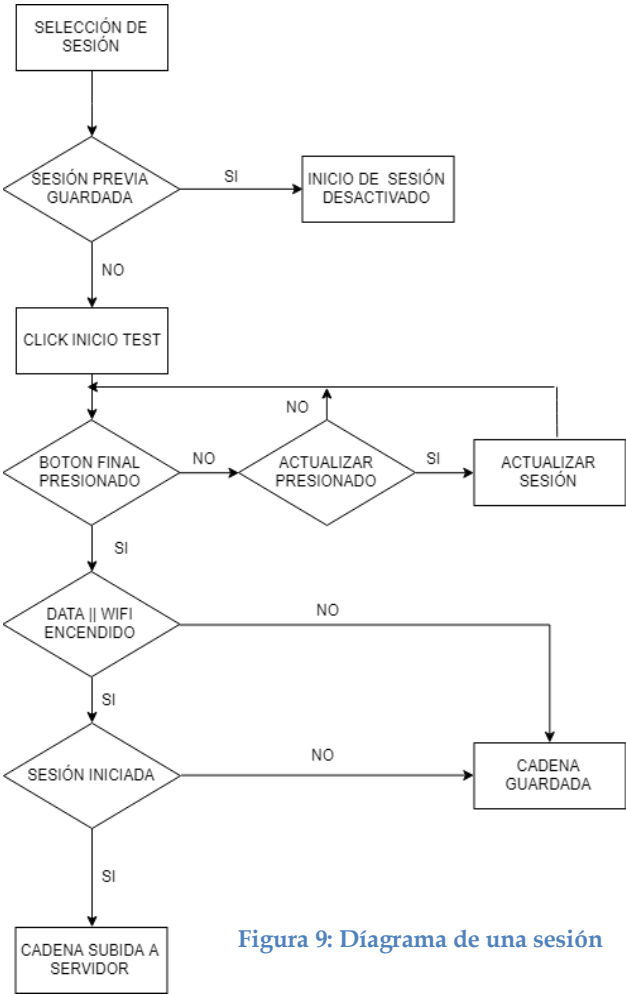


Figura 9: Diagrama de una sesión

5.2.6 Login y subida de datos

Para todas las comunicaciones realizadas entre el servidor y el móvil se ha utilizado la librería abierta Retrofit [7] junto a un Gson converter para trabajar con los datos JSON que llegan del servidor.

Para realizar el login un usuario, deberá introducir sus datos en la pantalla correspondiente, si el hash de la contraseña coincide con la del usuario guardado en la base de datos, se enviarian por un JSON los datos relevantes del usuario al móvil como por ejemplo su ID o nombre.

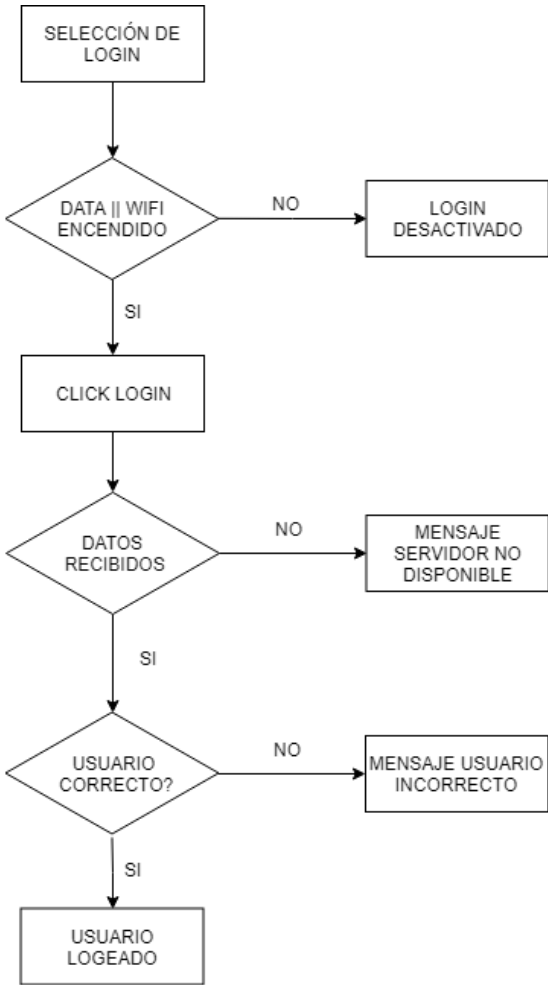


Figura 10: Diagrama de login

En caso de querer subir los datos, el usuario una vez logeado se podrá dirigir a la pantalla de subida de datos. En ella puede subir, si dispone de conexión a la red, y eliminar la sesión previa en caso que sea necesario.

5.2.7 Navegación entre actividades

Con tal de facilitar la claridad y navegación entre diferentes actividades se ha implementado un `NavigationDrawer` complementado con fragmentos de Android. De esta manera cada actividad es un fragmento que cuando seleccionado reemplaza el anterior fragmento que había colocado.

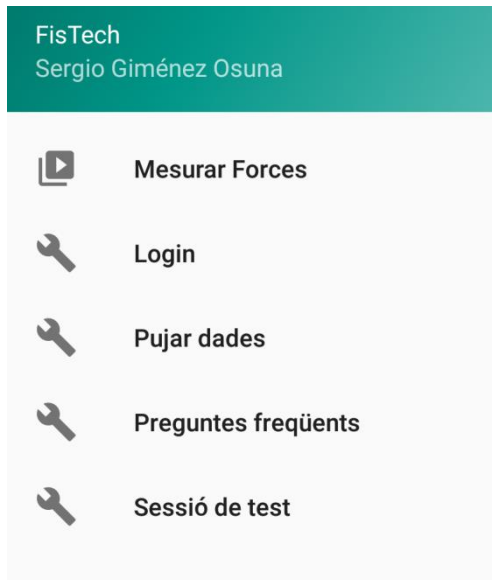


Figura 11: Navigation Drawer

5.2.7 Avisos y FAQ

Debido a que el sistema cuenta con restricciones para el uso de este, las cuales pueden ser no claras para el usuario se ha implementado una combinación de iconos codificados por colores para facilitar la resolución de estas. Además se ha dedicado una página explicando estos iconos y otras preguntas frecuentes que pueda tener.

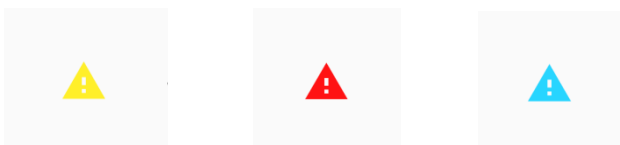


Figura 12: Alertas de restricciones

En la figura 12 podemos ver tres distintas alertas:

- **Amarilla:** El usuario no está logeado y no podrá subir la cadena de sesión.
- **Roja:** El usuario no dispone de conexión y no podrá logearse o subir una cadena.
- **Azul:** El usuario tiene una cadena previa guardada y no puede iniciar una nueva sesión.

5.3 Aplicación web

5.3.1 Página web

Con tal que el especialista pueda consultar las sesiones del usuario se ha creado una página web con todos los usuarios, las sesiones que estos han realizado y detalles sobre estas sesiones. La página ha sido implementada siguiendo un modelo MVC

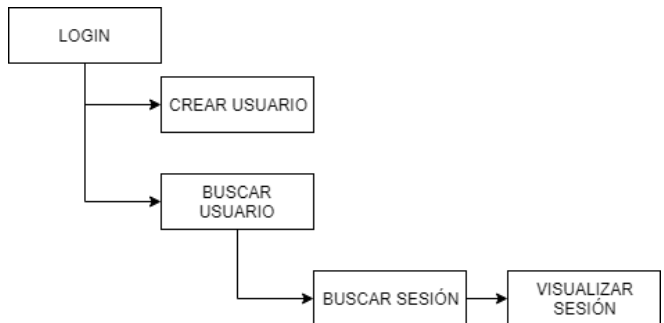


Figura 13: Mapa de la página

Como se puede ver en la Figura 13, se ha relegado la tarea de la creación del usuario al especialista, de manera que solo las personas autorizadas puedan acceder a las funcionalidades del guante.

El acceso a la página ha sido restringido a especialistas/desarrolladores, una vez estos hayan entrado correctamente a la página web podrán acceder a todos los módulos de esta. Siguiendo la ruta para acceder a una sesión, se pasará a la página de búsqueda de usuario, a esta se le ha añadido un campo de búsqueda de usuario con tal de agilizar la tarea una vez el nombre de usuarios sea considerablemente grande. Una vez seleccionado el usuario se mostrarán todas sus sesiones y se escogerá una de ellas.

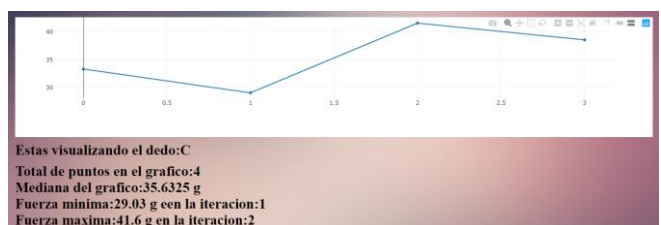


Figura 14: Página de visualización de sesión

Como se puede observar en la figura 14 se ha incluido un gráfico con las fuerzas realizadas en cada punto. Para crear este gráfico se ha hecho uso de la librería libre Plotly [8] escrita en JavaScript. También se han añadido datos como por ejemplo las repeticiones realizadas, la mediana y fuerzas máximas/mínimas en algún punto de la sesión.

5.3.2 Base de datos

La base de datos ha sido diseñada para ser lo mas simple cubriendo a la vez todas las necesidades que requiera la aplicación móvil, de esta manera cualquier expansión de la aplicación o base de datos pueda ser realizada de manera simple. Esta ha sido desarrollada en MySQL.

Tabla usuario

Compuesta de 5 filas: Id, nombre, contraseña, usuario y admin.

ID [Int]: Identifica de manera única cada fila de la tabla.

Nombre [Varchar]: Nombre real de la persona a la cual se asigna cada usuario.

Contraseña [Varchar]: Forma de autenticación del usuario.

Usuario [Varchar]: Nombre del usuario que se utilizara para identificarse en la aplicación.

Admin [Int]: Determina si el usuario es un especialista/ desarrollador o un paciente.

Tabla sesión

ID [Int]: Identifica de manera única cada fila de la tabla.

CadenaSesion [Varchar]: Guarda la cadena de todas las fuerzas realizadas por el usuario.

Día [Varchar]: Guarda el día en la que se finalizo la sesión.

Hora [Varchar]: Guarda la hora en la que se finalizo la sesión.

User_fk [Int]: Relaciona la tabla sesión con la tabla usuario.

5.3.3 Scripts de respuesta

Toda petición realizada desde la aplicación móvil mediante Retrofit es una llamada a un script escrito en PHP.

Se han requerido de 2 scripts distintos para realizar todas las funciones necesarias. Uno para realizar el login y otro para guardar la cadena de una sesión. Todos los datos son enviados mediante POST.

En el script preparado para el login se recogen tanto el usuario como la contraseña enviados por la petición POST del móvil, una vez recogidos se comprueban con la base de datos que estos sean iguales a los guardados. Finalmente, dependiendo de si la consulta SQL ha retornado los datos correctamente enviaremos un JSON con la información del usuario o uno con datos preparados los cuales denotan que el usuario se ha equivocado. Para el guardado la cadena el script se limita a recoger los datos relevantes de la petición post y insertarlos en la base de datos mediante una query de INSERT.

5.3.4 Extracción de fuerzas de una cadena

Debido a que las múltiples subcadenas que componen una sesión son guardadas al final como la concatenación de estas es necesario crear un sistema que sea capaz de aislar cada una de las fuerzas realizadas en una sesión que tiene asignado un dedo.

Para realizar esta tarea se requieren de varios elementos previos:

- Una cadena de sesión seleccionada
- Un array “mapa” con todas las letras existentes en la cadena[Pulgar, Indice, Corazon, Anular, Meñique, Final]
- Una letra de la cual aislar las fuerzas

En nuestro caso, si ninguna letra es seleccionada escogemos por defecto la P.

El primer paso a realizar es identificar cual será la letra final con la que poder calcular los punteros. Si por ejemplo la letra es la P la letra final será la I, en caso de ser la M escogeríamos la F.

Una vez guardadas las letras, deberemos buscar las primeras instancias de cada una de estas en el array de fuerzas y extraer la subcadena, esta siendo el resultado de la resta del puntero de la letra final con el inicial. En un nuevo array añadimos esta fuerza y recalculamos las nuevas posiciones. Debido a que PHP dispone solo de la función *strpos()* la cual retorna la primera instancia que encuentra de la cadena se debe eliminar la toda la subcadena de la que hemos recogido el dato.

Para realizar esto, se han seleccionado con la anterior función *strpos()* la primera letra “P” y la ultima letra que siempre se encuentra en la cadena “F” y es elimina toda la subcadena de la primera iteración.

Una vez hecho este proceso, se repetirá hasta que uno de los punteros retorne *null*, lo cual indica que ya no quedan mas fuerzas a procesar.

5.3.5 Creación de usuarios

Con tal que un fisioterapeuta pueda añadir usuarios el momento que estos son añadidos a un plan de rehabilitación, se ha incluido una página con un formulario que permite realizar esta acción.

En este también se permite añadir otros fisioterapeutas o administradores dependiendo si en la creación se pone un 0/1 en el campo de Admin.

6 Consideraciones de diseño

En este apartado se explicaran ciertas decisiones de diseño que condicionaron como se desarrollo el proyecto.

6.1 Información en una cadena

Inicialmente existieron dos posibles soluciones sobre como enviar todas las fuerzas de cada dedo, desde el microcontrolador al móvil y del móvil al guante.

Una era la cadena mencionada en este informe y otra era el envío de cada una de las fuerzas por separado.

Cada versión tenía sus pros y contras:

La cadena unificada de fuerzas permitía el hecho de solo enviar un objeto por el bluetooth simplificando el diseño del envío por Bluetooth, sin embargo, usar esta técnica tenía la contra de tener que implementar un parser cada vez que quiera ser tratada o información necesite ser extraída.

El envío separado de fuerzas permitía eliminar la necesidad de procesar las fuerzas ya que estas serian clasificadas el momento que fueran recibidas en cualquiera de los lugares que sean enviadas, sin embargo, esto implicaba una mayor complejidad, debido a que cada letra debía ser enviada y tratada Independientemente, aparte de implicar un mayor ratio de posibles fallos.

Finalmente se decanto por la versión de cadena unificada debido al desconocimiento inicial de las librerías utilizadas para la comunicación Bluetooth de datos desde el microcontrolador y el hecho que procesar la cadena no era una tarea compleja de por si.

6.2 Posibilidad de hacer una sesión desconectado

Inicialmente en el proyecto se iba a requerir que el usuario se identificase de manera inmediata antes de poder acceder a cualquiera de las funciones de la aplicación. Sin embargo la idea de poder guardar sesiones cambio la lógica de como el usuario se podía mover en la aplicación.

Por tanto, se paso a un diseño en el que el usuario tendría acceso a todas las funciones de la aplicación, pero una vez realizada la sesión se restringiría el poder realizar una sesión hasta que no se identificase.

7 Problemas en el proyecto

A continuación se mostraran ciertos contratiempos que surgieron mientras el proyecto era diseñado y como se afrontaron.

7.1 Protocolo de envío automático

Cuando se empezó a implementar la comunicación Bluetooth entre móvil y guante, primero se empezó con un botón en la aplicación el cual envía una señal o ping para señalar al guante que este quería las fuerzas.

Posteriormente se paso a implementar un protocolo el cual pudiera hacer esta tarea sin necesidad de un botón.

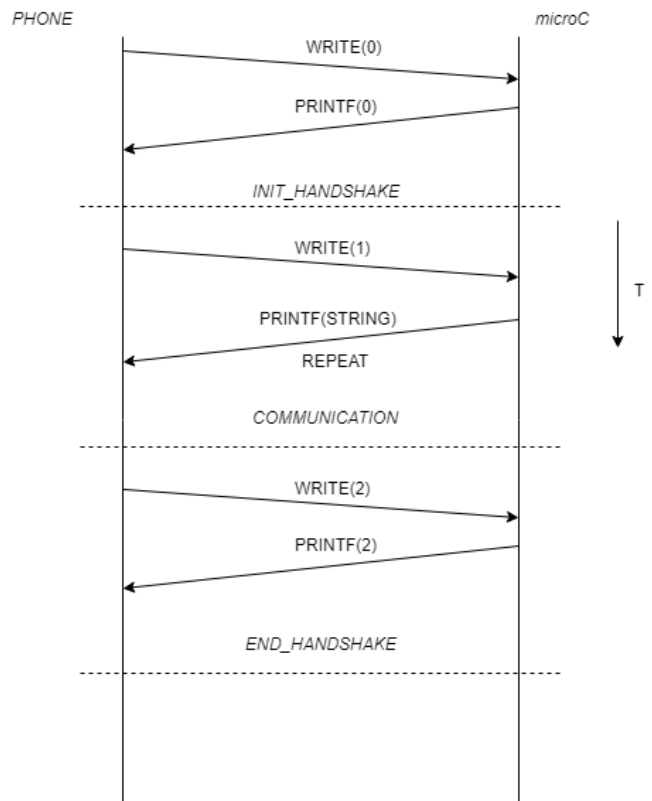


Figura 15: Protocolo de comunicación

Como se puede ver en la figura, la comunicación se inicia mediante un *handshake*, asegurando que el microcontrolador esta preparado. Una vez pasado esto el móvil pasaría a comenzar a pedir datos repetidamente hasta que el usuario finalizase la sesión, la aplicación enviara la señal de final al microcontrolador y este indica al móvil que ha finalizado el envío de datos.

Sin embargo múltiples problemas surgieron en la fase de comunicación, el problema se daba en la forma de un crash momentos después de iniciarse el protocolo.

Múltiples posibles errores fueron barajados:

- El microcontrolador no podía procesar tantas peticiones.
- El bufer que mantiene el móvil se llena demasiado rápido debido a que no puede procesar todas las peticiones.

Con tal de poder solucionar ambos problemas a la vez se probó a implementar una parada asíncrona de manera que el microcontrolador tuviera tiempo a procesar todas las peticiones y el móvil pudiese vaciar el bufer a tiempo.

Esta solución no funcionó y finalmente se desechó la idea del protocolo por la del botón la cual es más intuitiva que el hecho de un protocolo por el cual no tienen ningún control sobre.

7.2 Gráfico de visualización de sesión

Inicialmente se buscó una librería libre que pudiera dibujar un gráfico de línea recta sobre dos ejes X, Y en PHP, debido a que todo el procesamiento de los datos se hacía en este lenguaje.

Sin embargo todas las librerías encontradas requerían de pago, estaban desactualizadas o tenían capacidades muy limitadas. Por esto, se hizo la concesión de utilizar librerías que usan JavaScript a cambio de un código más complejo debido a la intercalación necesaria de ambos lenguajes.

8 Líneas de mejora futuras

- Mantener varias sesiones: Mientras que en la actualidad solo se mantiene una sesión, es posible que en el futuro sea una buena idea permitir crear múltiples sesiones cuando el usuario no dispone de conexión a internet.
- Uso de nueva librería gráfica: Como ya se ha mencionado anteriormente, la librería actual tiene ciertas limitaciones a la hora de mostrar la información, por lo tanto buscar un reemplazo mientras no imprescindible sería una mejora.
- Mejora del aspecto de la web: La página web ha sido actualmente diseñada con la finalidad de mostrar los datos y se ha priorizado función sobre forma. Mejorar el aspecto y CSS sería ideal para facilitar la navegación del especialista.
- Grafos en el móvil: Otra posible línea de mejora sería visualizar en un grafo semejante al de la web las fuerzas de las sesiones anteriores o de la guardada en el móvil.
- Actualizar los sensores: Actualmente los sensores captan fuerzas cuando ninguna presión se le somete y el sensor del dedo índice siempre capta alguna lectura. Sería necesario reemplazar todos los sensores con tal de conseguir siempre lecturas correctas y evitar la necesidad de implementar filtros para lecturas falsas.

9 Conclusiones

Mientras que se han asumido los objetivos iniciales del proyecto, este aún tiene un amplio potencial de mejora y desarrollo en todas las partes que lo componen.

10 Agradecimientos

Agradecimientos a Dolores Rexachs por darme soporte e ideas a lo largo del proyecto.

A Emili Isamat por su ayuda a lo largo del proyecto con dudas respecto al lenguaje Android.

Y a Roger Recasens por proporcionarme con toda la información necesaria para iniciar el proyecto.

11 Bibliografía

- [1] Roger Recasens, [Online] Disponible en <https://ddd.uab.cat/record/181581?ln=ca>
- [2] Jordi González, Sistema de adquisición y procesamiento de señales para aplicaciones biomédicas. Universitat Autònoma de Barcelona (UAB), 2014.
- [3] Ejemplo de guante mecánico, [Online] Disponible en <https://www.neofect.com/en/product/rapael/>
- [4] Ejemplo de guante electrónico, [Online] Disponible en <https://www.saebo.com/saeboglove>
- [5] EventBus, [Online] Disponible en <https://github.com/greenrobot/EventBus>
- [6] MPAndroidChart, [Online] Disponible en <https://github.com/PhilJay/MPAndroidChart>
- [7] Retrofit, [Online] Disponible en <https://square.github.io/retrofit/>
- [8] Plotly, [Online] Disponible en <https://plot.ly/>

