



MÁSTER EN SISTEMAS INTELIGENTES

TRABAJO DE FINAL DE MÁSTER

**Mejoras en el reconocimiento de actividades
con redes neuronales para terapias de salud
mental**

Autor:
Miguel Matey Sanz

Tutor:
Joaquín Huerta Guijarro
Cotutor:
Sven Casteleyn

Fecha de lectura: 25 de septiembre de 2020
Curso académico 2019/2020

Resumen

Hoy en día, gran cantidad de aplicaciones móviles emplean información contextual del usuario para su funcionamiento, como, por ejemplo, su ubicación o actividad. En el campo de la salud mental, esta información contextual es vital para el tratamiento de trastornos mentales, pero ésta no siempre puede obtenerse con los requisitos de precisión y frecuencia requeridos —los cuales pueden variar mientras se obtienen los datos— empleando los sistemas básicos de los dispositivos móviles.

En este documento se muestra el trabajo desarrollado en el contexto del proyecto SyM-ptOMS, el cual investiga la capacidad de los dispositivos móviles para ser utilizados con aplicaciones psicológicas para pacientes con problemas de salud mental. El trabajo ha consistido en el desarrollo de un sistema de detección de actividades sensible al contexto para la monitorización y seguimiento de pacientes con trastornos de salud mental mediante el *framework* de desarrollo multiplataforma NativeScript, aunque la implementación en sistemas iOS queda fuera del alcance de este trabajo. Este sistema permite emplear los mecanismos de reconocimiento proporcionados por el sistema operativo Android, los cuales trabajan a una determinada frecuencia de reconocimiento y con actividades limitadas, lo cual no es suficiente para la monitorización de los pacientes. Para solucionar esto, el sistema permite utilizar un sistema propio basado en una red neuronal, el cual presenta una mayor frecuencia de reconocimiento y es capaz de identificar cualquier tipo de actividad siempre que haya sido previamente entrenada. Además, el sistema es capaz de emplear estos sistemas en función del contexto del usuario, pudiendo ajustarse mejor a los requisitos de monitorización del momento.

El funcionamiento del modelo de predicción propio y del sistema se ha comprobado simulando un caso de uso de un paciente con agorafobia, con el objetivo de que el sistema seleccionase el mecanismo de reconocimiento más adecuado según la información contextual. Los resultados muestran como el modelo de predicción presenta buenos resultados tanto con datos de prueba como con datos reales, y como el sistema sensible al contexto emplea los distintos sistemas de reconocimiento en función del contexto del usuario según se ha configurado el sistema.

Palabras clave

Reconocimiento de actividades, conocimiento del contexto, redes neuronales, Android, NativeScript

Abstract

Nowadays, a large number of mobile applications use contextual information for their operation, such as user's location or activity. In the field of mental health, this contextual information is vital for the treatment of mental disorders. However, this information cannot always be obtained with the required precision and frequency requirements—which may vary while the data is being gathered—using the systems provided by the mobile devices.

This document shows the developed work in the context on the SyMptOMSproject, which investigates smartphones' capabilities to be used for psychological applications for patients with mental disorders. The performed work consists of the development of a context aware activity detection system for the tracking and monitoring of mental disorders patients. The system has been developed using the multiplatform development framework NativeScript, although the iOS implementation is out of the scope of the work. This system allows to use the provided recognition mechanisms by the Android operative system, which work at a determined recognition frequency and with limited activities. This is not enough for patient tracking. To solve this, the system allows to use an own neural network based system, which has a higher recognition frequency and is able to identify any kind of activity as long as it has been trained. Moreover, the system is able to use these systems based on user's context, adapting itself to the tracking requirements in each moment.

The performance of the own model and the system has been tested by simulating an use case of an agoraphobia patient, with the objective that the system employed the most suitable recognition mechanism based on contextual information. Results show that the prediction model has good results both with test data and with real data, and that the context aware activity detection system uses the recognizing systems based on the user's context according how the system has been configured.

Keywords

Human Activity Recognition, context awareness, neural networks, Android, NativeScript

Índice general

1. Introducción	7
1.1. Contexto	8
1.2. Objetivos y metodología	8
1.3. Estructura	9
2. Trabajo relacionado	11
2.1. Uso de datos de ubicación	11
2.2. Reconocimiento de actividades	11
2.3. Reconocimiento de actividades teniendo en cuenta el contexto	12
3. Requisitos y arquitectura del sistema	15
3.1. Requisitos	15
3.2. Arquitectura del sistema	17
4. Implementación del sistema	19
4.1. ActivityRecorder	19
4.2. Model Training	21
4.2.1. Extracción de características	22
4.2.2. Arquitectura y entrenamiento del modelo	23
4.3. Activity Recognition	25
4.3.1. Context APIs	25

4.3.2. Context Activity Recognition	27
5. Evaluación	31
5.1. Evaluación del modelo de reconocimiento propio	31
5.2. Evaluación de la selección de sistema de reconocimiento dependiendo del contexto	34
6. Conclusiones	37
Bibliografía	39

Capítulo 1

Introducción

Los sistemas sensibles al contexto (*context-aware systems*) son aquellos que emplean cualquier información (contexto) que pueda ser usada para caracterizar la situación y el entorno de un usuario con el objetivo de proporcionarle información o servicios relevantes [14]. La información de contexto más empleada habitualmente es la localización, utilizada por una gran cantidad de investigadores y sistemas en campos como la salud [5]. Un claro ejemplo de un sistema que emplea la información contextual de la ubicación de los usuarios son las aplicaciones del tiempo en dispositivos móviles, las cuales muestran al usuario el tiempo de la zona donde este se encuentre. Otro contexto menos empleado que la localización es la actividad que realiza el usuario. La utilización de este tipo de información puede ser empleado en diversas aplicaciones, por ejemplo, la identificación de situaciones de emergencia, o la caída de personas con avanzada edad [17].

En el campo de la salud mental, tanto la localización como los movimientos y actividades del paciente se consideran recursos clave para el tratamiento de trastornos mentales como la depresión [32][31]. Esta información de contexto puede obtenerse mediante el dispositivo móvil y emplearse para evaluar el comportamiento del paciente tanto previamente como en el momento de realizar una *ecological momentary intervention* (EMI) [27], un tipo de tratamiento realizado *in-situ* (lugar de interés para el trastorno que sufre el paciente) y en tiempo real.

En estos casos, la principal dificultad reside en la detección de la actividad del paciente. En el caso de los dispositivos móviles Android, el propio sistema operativo proporciona dos servicios para llevar a cabo esta detección. El primero de ellos permite detectar transiciones entre una actividad y otra, siendo las actividades posibles a detectar muy limitadas. El segundo servicio proporcionado por Android permite detectar la actividad que realiza el usuario cada cierto intervalo de tiempo, el cual es mayor que el servicio anterior y que presenta una menor precisión. Al igual que con el primer servicio, las actividades a detectar están muy limitadas. Este hecho hace que emplear estos servicios pueda ser útil o no, ya que dependiendo del trastorno del paciente y de su localización se necesitarán detectar unas actividades específicas durante las EMI, posiblemente no soportadas por dichos servicios. Por ejemplo, emplear el primer servicio indicado puede ser útil para conocer cuando el paciente ha empezado o dejado de conducir, pero no resulta de utilidad en el momento de realizar una EMI para proporcionar *feedback* al terapeuta sobre las actividades específicas relacionadas con el trastorno del paciente.

Por lo tanto, la detección de dichas actividades específicas necesita de otro tipo de sistema más adecuado para este contexto de aplicación. En concreto, se necesitaría un sistema de detec-

ción de actividades personalizado con una mayor frecuencia de reconocimiento de actividades y que permita detectar cualquier actividad (tras un determinado aprendizaje), empleando ciertos métodos y técnicas de la literatura. Por lo que respecta a los métodos de detección de actividades, los más empleados son aquellos que emplean acelerómetros, magnetómetros o giroscopios, aunque la detección puede llevarse a cabo empleando tan solo uno de ellos [11]. Por ejemplo, en [20] se emplean los datos en crudo del acelerómetro del dispositivo móvil para detectar unas determinadas actividades, comparando los resultados obtenidos a partir de regresión logística, árboles de decisión y redes neuronales.

1.1. Contexto

El contexto en el que se enmarca este trabajo es dentro del proyecto de I+D nacional SyMptOMS¹, desarrollado en el grupo de investigación GEOTEC, en colaboración con el grupo de investigación LABPSITEC. En SyMptOMS se investiga cómo las capacidades de los dispositivos móviles (GPS, sensores, detección de entorno y contexto, etc.) y de otros dispositivos *wearables* (relojes inteligentes, pulseras de actividad, etc.) pueden emplearse para desarrollar nuevos métodos de evaluación e intervención psicológica para pacientes con problemas de salud mental.

Actualmente, SyMptOMS está compuesto por una aplicación móvil para la recogida de datos y realización de intervenciones, y por una aplicación web que permite configurar la aplicación móvil para indicar qué datos se deben recoger, qué análisis realizar, qué intervenciones efectuar, y qué visualizaciones obtener a partir de los datos de los pacientes.

Con el objetivo de mejorar el proceso de recogida de datos y de realización de intervenciones, se quiere integrar un sistema para la monitorización de las actividades de los pacientes durante su día a día y durante la realización de intervenciones terapéuticas, adaptándose a ambos contextos.

1.2. Objetivos y metodología

En este proyecto se busca diseñar un sistema de detección de actividades sensible al contexto que emplee la ubicación del usuario para determinar qué sistema (o modelo) de detección de actividades disponible es adecuado en cada situación. Estos modelos disponibles serán ambos servicios de detección que proporciona Android y un modelo propio para la detección de actividades comunes y específicas. Basándonos en la literatura, el modelo propio propuesto consistirá en una red neuronal que hará uso de ciertas características extraídas de datos del acelerómetro del dispositivo móvil para realizar la detección. La creación del modelo propio implicará la recogida de datos para el entrenamiento y la validación de la precisión del mismo. Así pues, los principales objetivos del proyecto son los siguientes:

- **O1:** Diseñar y desarrollar un modelo propio de detección de actividades comunes y específicas para pacientes con trastornos de salud mental.
- **O2:** Desarrollar un sistema de detección de actividades sensible al contexto, el cual seleccione el sistema de reconocimiento a emplear en función del contexto del usuario.

¹“Retos Investigación” 2018, RTI2018-099939-B-I00. <http://geotec.uji.es/projects/SyMptOMS/>

La metodología adoptada en este proyecto consistirá en una metodología de trabajo ágil, desarrollando de forma incremental cada uno de los componentes de los que depende el sistema propuesto. Esta metodología permitirá desarrollar poco a poco las funcionalidades del sistema propuesto y probar su correcto funcionamiento.

Para llevar a cabo el objetivo O1 se crearán dos componentes que formarán parte del sistema propuesto. El primero de ellos será una aplicación Android para la recogida de datos del acelerómetro asociados a determinadas actividades. Estos datos se emplearán para el entrenamiento del modelo propio. El otro componente consistirá en el uso de un conjunto de servicios en la nube que se emplearán para el tratamiento de los datos de entrenamiento recogidos y el propio entrenamiento de la red.

Para la realización del objetivo O2 se crearán otros dos componentes, los cuales serán desarrollados mediante el *framework* de desarrollo multiplataforma NativeScript², ya que es el lenguaje empleado actualmente en el proyecto SyMptOMS. Aún así, cabe destacar que el sistema propuesto solo se implementará para dispositivos Android, quedando fuera del alcance del proyecto la implementación para dispositivos iOS. El primero de estos componentes será un *plugin* que se encargue de la gestión y utilización de los sistemas de reconocimiento y de proporcionar actualizaciones sobre la ubicación del dispositivo. El segundo componente será una aplicación demostradora para un caso de uso que hará uso del componente anterior junto con técnicas de *geofencing* para crear el sistema de detección de actividades sensible al contexto propuesto.

1.3. Estructura

De aquí en adelante, el presente documento se estructura en los siguientes capítulos:

- **Capítulo 2 – Trabajo relacionado:** Se presentan conceptos y técnicas de la literatura sobre los que se cimienta el presente trabajo.
- **Capítulo 3 – Requisitos y arquitectura del sistema:** Se especifican los requisitos del sistema propuesto y se proporciona una breve visión sobre la arquitectura del sistema a alto nivel.
- **Capítulo 4 – Implementación del sistema:** Se detalla el funcionamiento y los aspectos más relevantes de cada uno de los componentes de la arquitectura del sistema.
- **Capítulo 5 – Evaluación:** Se evalúa el rendimiento del modelo de predicción propio y del sistema propuesto.
- **Capítulo 6 – Conclusiones:** Discusión de los resultados obtenidos y trabajo futuro.

²Native mobile apps with Angular, Vue, TypeScript, JavaScript – NativeScript

Capítulo 2

Trabajo relacionado

2.1. Uso de datos de ubicación

Hoy en día, todos los dispositivos móviles y algunos dispositivos *wearables* disponen de un sensor GPS, el cual permite conocer la ubicación del dispositivo en un momento determinado. Realizando un seguimiento de la ubicación del dispositivo se puede conocer la ruta seguida por el usuario del dispositivo. Además, si se recoge otro tipo de información, ya sea proveniente de aplicaciones o de otros sensores del dispositivo, se puede conseguir información contextualizada que nos permite detectar sucesos en el espacio y el tiempo.

Una de las utilidades que pueden darse a los datos de ubicación es determinar cuándo el usuario del dispositivo móvil entra o sale de un determinado lugar. Este lugar viene definido como un punto de coordenadas GPS y un radio, lo cual crearía un perímetro virtual, también llamado *geofence* (geovalla). Con este *geofence*, la ubicación del dispositivo móvil y matemáticas básicas se puede determinar si el dispositivo está dentro o fuera de este área virtual, y si entra o sale del mismo. A este proceso se le denomina *geofencing* (geovallado).

2.2. Reconocimiento de actividades

El reconocimiento de actividades es un área de estudio que se basa en el uso de diversos tipos de sensores cercanos (p.e. móvil) o sujetos (p.e. reloj inteligente) al cuerpo humano con el objetivo de predecir la actividad que está realizando el usuario. Comúnmente, suele emplearse un único dispositivo de sensorización, que normalmente utiliza acelerómetros, magnetómetros y/o giroscopios, aunque en algunos artículos se emplean varios dispositivos [11]. Las actividades detectadas suelen ser bastante comunes, como caminar, correr, estar sentado o subir y bajar escaleras [1][20]. Por otro lado, existen trabajos que se centran en la detección de actividades específicas a ciertos ámbitos, como, por ejemplo, el deporte, detectando actividades como sentadillas y flexiones [26] o aerobic [3].

En cuanto a la localización de los dispositivos de sensorización, ésta suele ser muy diversa, pudiendo emplearse en el pecho [10][25], en la cintura [23][35], en los brazos [30], muñecas [8][24], muslos [38][16][22] o los tobillos [9]. De todas las posibles localizaciones, según Bao y Intille [2], el muslo es la que permite obtener mejores resultados, al menos, para el conjunto de actividades

seleccionado por los autores.

Pese a una revisión reciente de Cornacchia y otros [11], en la que afirman que el uso del acelerómetro como único sensor para llevar a cabo la detección de actividades no es suficiente y debe acompañarse de otros acelerómetros u otros tipos de sensores, en la literatura aparece gran variedad de artículos donde tan solo se emplea un acelerómetro como sensor y donde se obtienen resultados bastante satisfactorios [3, 13, 20, 26, 36].

Por otro lado, Figo y otros [15] estudian y clasifican una amplia diversidad de técnicas existentes para la extracción de características a partir de las series temporales obtenidas de un acelerómetro. Además, los autores realizan un estudio del coste temporal y espacial de cada uno de los métodos, indicando su viabilidad para su ejecución en dispositivos móviles o con recursos limitados. En cuanto al proceso de extracción de características, la literatura coincide en emplear la técnica *sliding window with overlapping*, la cual divide la señal en subconjuntos superpuestos sobre los que se extraen las características [3, 12, 13, 20, 28, 29, 36].

Por último, en lo respectivo a los posibles métodos de inferencia o clasificadores a emplear, un estudio de Kwapisz [20] compara el funcionamiento de un árbol de decisión, regresión logística y una red neuronal y sus resultados muestran como cada clasificador destaca en un determinado tipo de actividad, siendo la red neuronal el mejor clasificador de media. También, Bayat y otros [3] realizan una comparación con una red neuronal, un RF (*Random Forest*¹), una SVM (*Support Vector Machine*²) y un *LogitBoost*³, entre otros, obteniendo los mejores resultados con la red neuronal. En un estudio similar, Sukor y otros [36] realizan una comparativa entre una red neuronal, un árbol de decisión y una SVM, obteniendo la red neuronal los mejores resultados.

2.3. Reconocimiento de actividades teniendo en cuenta el contexto

Los sistemas de reconocimiento o detección de actividades sensibles al contexto son aquellos que emplean la información contextual del usuario para facilitar y/o complementar el reconocimiento de la actividad que desarrolla el usuario. Actualmente, con las capacidades de los dispositivos móviles y su gran cantidad de sensores, el reconocimiento de actividades teniendo en cuenta el contexto del usuario es cada vez más sencillo. Por ejemplo, Cao y otros [6] proponen un método de detección de actividades empleando las actividades detectadas con anterioridad como información de contexto para validar y corregir, en caso necesario, las predicciones de actividades actuales.

Por otro lado, no es necesario emplear un dispositivo móvil para crear sistemas de este tipo, pudiendo emplearse dispositivos de IoT (*Internet of Things*) como los de *Smart Homes* u otros recursos. En [37], los autores proponen un sistema para la detección de actividades propias del hogar (ver la televisión, trabajar en el ordenador, comer, cocinar, etc.) en una *Smart Home*, teniendo en cuenta la información contextual proporcionada por los sensores de la casa (localización en el hogar, dispositivos electrónicos en uso, etc.). En otro ámbito, Zhu y otros [39] proponen un *framework* matemático para la utilización de información contextual en vídeo (relación espacio-temporal entre objetos) para el reconocimiento de actividades.

¹*Random Forest* – Wikipedia

²*Support Vector Machine* – Wikipedia

³*LogitBoost* – Wikipedia

Este tipo de sistemas también se emplean en la monitorización de pacientes que sufren algún tipo de enfermedad, como por ejemplo para la detección de actividades para pacientes con Alzheimer [19]. Dicho sistema se basa en la información contextual de ubicación, cámaras y dispositivos *wearables* para detectar actividades de alto nivel de interés en este tipo de enfermedades, como, por ejemplo, tomar la medicación.

Capítulo 3

Requisitos y arquitectura del sistema

En este capítulo se recogen los requisitos funcionales que debe cumplir el sistema de detección de actividades sensible al contexto, así como la arquitectura de alto nivel propuesta para el sistema.

3.1. Requisitos

En esta sección se muestran los requisitos que debe cumplir el sistema de detección de actividades propuesto. Dichos requisitos han sido generados por los miembros del proyecto SyMptOMS y los terapeutas del grupo LABPSITEC, mediante un estudio sistemático de la literatura y teniendo en cuenta las necesidades del proyecto y sus pacientes.

Como ya se ha comentado anteriormente, existen distintas formas de llevar a cabo la detección de actividades mediante un dispositivo móvil Android. Este sistema operativo proporciona a los desarrolladores una API¹ para la detección de ciertas actividades (ver Tabla 3.1) mediante dos sistemas:

- **ActivityTransition**: detección de cambios entre actividades con una gran precisión, pero con un considerable retardo (ej. pasar de ir en coche a caminar, de caminar a correr...).
- **ActivityUpdate**: detección de actualizaciones en las actividades con una precisión y retardo menores que el sistema anterior. De hecho, el sistema anterior emplea este para realizar la detección de cambios entre actividades.

El hecho de que estos sistemas tan solo detecten un determinado conjunto limitado de actividades implica que si el desarrollador está interesado en detectar otro tipo de actividades se vea obligado a implementar su propio sistema de reconocimiento. Como ocurre en el proyecto de investigación en el que se desarrolla este trabajo, las actividades que pueden reconocer los sistemas de Android son de interés pero además es necesario reconocer otro tipo de actividades relacionadas con el ámbito del proyecto. Por lo tanto, se emplearán ambos sistemas de Android y se implementará un sistema de reconocimiento propio para un determinado conjunto de actividades específicas. Dicho sistema, el cual se basará en una red neuronal, se deberá entrenar

¹ActivityRecognitionClient

Actividad	ActivityTransition	ActivityUpdates
En vehículo	Sí	Sí
Correr	Sí	Sí
Caminar	Sí	Sí
En bicicleta	Sí	Sí
De pie	Sí	Sí
Inclinación	No	Sí
Desconocido	No	Sí

Tabla 3.1: Actividades reconocibles por los sistemas Android

con los datos de acelerómetro capturados durante la realización de las actividades a reconocer. Además se tendrá la posibilidad de reentrenar el sistema con datos obtenidos de nuevos pacientes con el objetivo de adaptar el sistema a estos nuevos pacientes.

Dentro del contexto del proyecto SyMptOMS, estos sistemas se emplearán para monitorizar a pacientes con problemas de salud mental, aunque no siempre los requisitos de monitorización son los mismos. Por ejemplo, cuando el paciente se desplaza de su hogar al trabajo en coche, no se requiere una frecuencia de monitorización elevada, siendo más interesante saber cuando el paciente ha empezado y acabado el trayecto. Por otro lado, mientras el paciente se somete a un proceso de exposición terapéutica (ej. un paciente con agorafobia en un lugar abierto) sí que sería necesaria una frecuencia de monitorización mayor, pudiendo detectar al instante comportamientos y actividades que puedan indicar que el paciente ha entrado en pánico.

Así pues, los sistemas de detección de actividades mencionados se emplearían en los siguientes casos:

- Cuando el paciente se encuentre haciendo vida normal o en lugares donde no se requiere su monitorización continua, siendo solo necesario saber que actividad está o ha estado realizando, se empleará el sistema **ActivityTransition**.
- Cuando el paciente se encuentre en algún lugar de interés relacionado con su trastorno de salud mental, requiriendo una mayor frecuencia de monitorización, se empleará el sistema **ActivityUpdate**.
- Cuando mediante el sistema **ActivityUpdate** se detecte un determinado tipo de actividad que pueda sugerir la necesidad de una mayor frecuencia de monitorización, se empleará el sistema de reconocimiento propio. Por ejemplo, el hecho de que un paciente con agorafobia se quede parado durante una terapia de exposición puede ser el detonante para que se active el sistema de reconocimiento propio.

Para llevar esto a cabo, el sistema necesitará conocer la localización del usuario cada cierto periodo de tiempo con el objetivo de detectar si el paciente se encuentra o no en algún lugar de interés (ej. espacios abiertos en pacientes con agorafobia) y activar o desactivar los sistemas de detección correspondientes.

Por tanto, los requisitos funcionales del sistema de detección de actividades sensible al contexto son:

- **RF1:** El sistema debe obtener periódicamente la localización del paciente.

- **RF2:** El sistema debe detectar cambios entre actividades empleando el sistema de reconocimiento de actividades `ActivityTransition`.
- **RF3:** El sistema debe detectar actualizaciones de actividades empleando el sistema de reconocimiento de actividades `ActivityUpdates`.
- **RF4:** El sistema debe detectar la actividad del paciente empleando el sistema propio de reconocimiento de actividades.
- **RF5:** El sistema debe emplear el sistema de reconocimiento de actividades adecuado dependiendo del contexto del paciente.
- **RF6:** El sistema debe permitir la captura de datos de acelerómetro para el entrenamiento del sistema de reconocimiento propio.
- **RF7:** El sistema debe permitir reentrenar el sistema de reconocimiento propio con nuevos datos de entrenamiento.

Además de los requisitos funcionales, se han definido otros requisitos centrados en el rendimiento del sistema de reconocimiento de actividades propio:

- **RR1:** El sistema de reconocimiento de actividades propio debe tener una precisión adecuada en la identificación de las actividades.
- **RR2:** El sistema de reconocimiento de actividades propio debe añadir al sistema el mínimo sobre coste temporal posible.

3.2. Arquitectura del sistema

La visión general de la arquitectura a alto nivel del sistema de detección de actividades sensible al contexto propuesto puede verse en la Figura 3.1. En esta arquitectura pueden distinguirse los siguientes módulos y sus correspondientes componentes:

- **Activity Recording:** módulo para la recogida de registros de acelerómetros asociados a una determinada actividad. Está formado por el componente `ActivityRecorder`, una aplicación Android encargada de dicha recogida de datos. Dichos registros serán empleados para el entrenamiento del modelo de reconocimiento de actividades propio.
- **Model Training:** componente que emplea los Servicios de Google Cloud Platform, donde Google ofrece una gran cantidad de productos² en diversos ámbitos, como en almacenamiento de datos, computación *serverless* o inteligencia artificial. Algunos de estos servicios se emplean para la extracción de características a partir de los datos en crudo del acelerómetro o para el entrenamiento del modelo de red neuronal para el reconocimiento de actividades.
- **Activity Recognition:** módulo para el reconocimiento de actividades. Está formado por dos componentes: `Context APIs` y `Context Activity Recognition`. Por una parte,

²Products & Services | Google Cloud

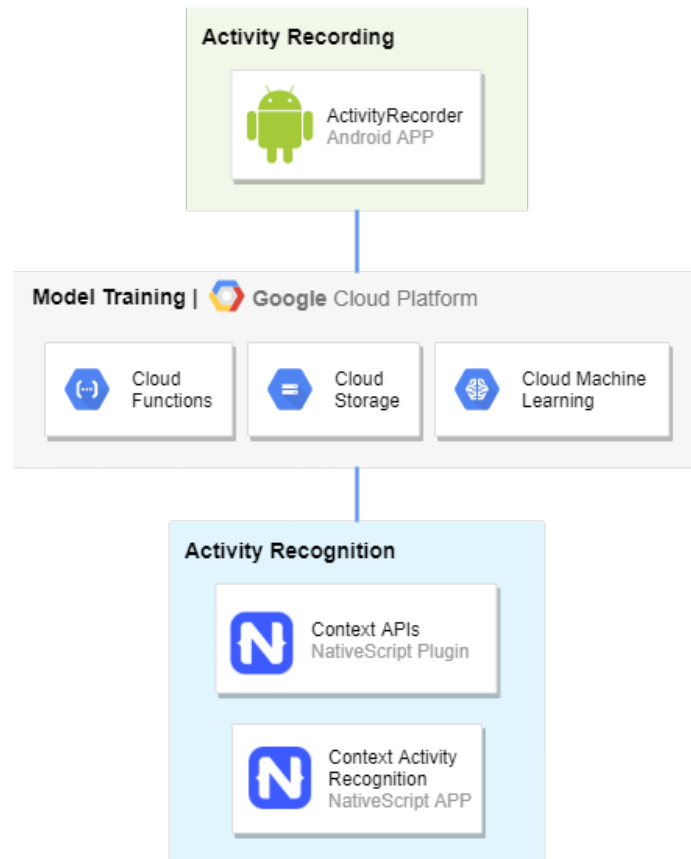


Figura 3.1: Arquitectura de alto nivel del sistema propuesto.

`Context APIs` es un *plugin* desarrollado mediante el *framework* de desarrollo multiplataforma NativeScript, el cual se encarga de proporcionar la geolocalización del usuario y el reconocimiento de actividades, permitiendo seleccionar el sistema de reconocimiento a emplear (`ActivityTransition`, `ActivityUpdates` o sistema propio). Por otro lado, `Context Activity Recognition` es una aplicación que hace uso del *plugin* anterior y que se encarga de seleccionar los sistemas de reconocimiento a emplear dependiendo del contexto del usuario (tanto ubicación como actividad actual).

En el siguiente capítulo se explicarán algunos de los detalles de implementación más relevantes de estos componentes, así como la interacción entre los mismos.

Capítulo 4

Implementación del sistema

El presente capítulo describe los detalles de implementación de cada uno de los componentes y como interactúan entre sí para lograr el correcto funcionamiento del sistema propuesto de detección de actividades sensible al contexto.

4.1. ActivityRecorder

Este primer componente, una aplicación Android, permite la recogida de muestras del acelerómetro durante la realización de una determinada actividad para, posteriormente, ser empleadas en el entrenamiento del modelo de detección de actividades propio (requisito RF6). El dispositivo se situará en el bolsillo delantero izquierdo del pantalón, ya que, emplear el dispositivo sensor en el muslo permite obtener los mejores resultados [2]. Las actividades que pueden entrenarse son: estar quieto, caminar, correr, estar sentado y estar sentado en cuclillas. Esta última actividad puede representar un estado de miedo o pánico en el paciente durante una exposición terapéutica.

En la Figura 4.1 se muestra la arquitectura y el flujo de trabajo de la aplicación de captura desarrollada, y la Figura 4.2 muestra la interfaz gráfica de la misma. La aplicación permite seleccionar una de las actividades indicadas anteriormente a la cual van a pertenecer los registros de acelerómetro recogidos. Una vez seleccionada una actividad y el usuario presiona el botón “START”, la aplicación arranca un servicio en primer plano que solicita recibir actualizaciones del estado del acelerómetro con una frecuencia de 50Hz (20 milisegundos), según como se sugiere en la literatura [18][13]. Dichas actualizaciones son entregadas a un receptor, el cual va almacenándolas en la memoria dinámica. Cuando el usuario termina de realizar la actividad y presiona el botón “STOP”, el servicio obtiene los registros del receptor y los almacena localmente y en *Cloud Storage* en formato CSV¹.

Un detalle a tener en cuenta durante la recogida de datos es el tiempo que transcurre entre que el usuario presiona el botón “START” y guarda su dispositivo móvil en su bolsillo. Del mismo modo, también se debe tener en cuenta el tiempo que transcurre mientras el usuario saca el dispositivo de su bolsillo y presiona el botón “STOP”. Dichos movimientos podrían introducir ruido en los datos de entrenamiento y reducir la precisión del modelo de predicción

¹CSV: *Comma-separated values*

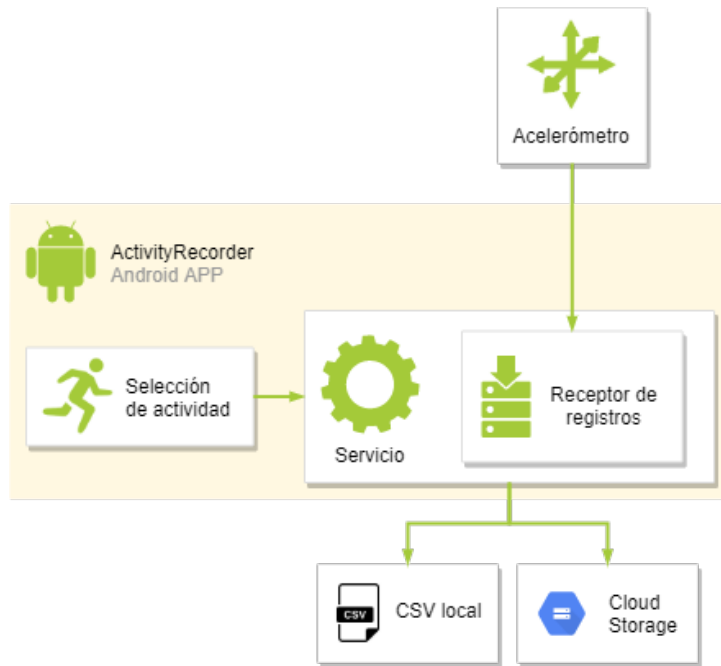
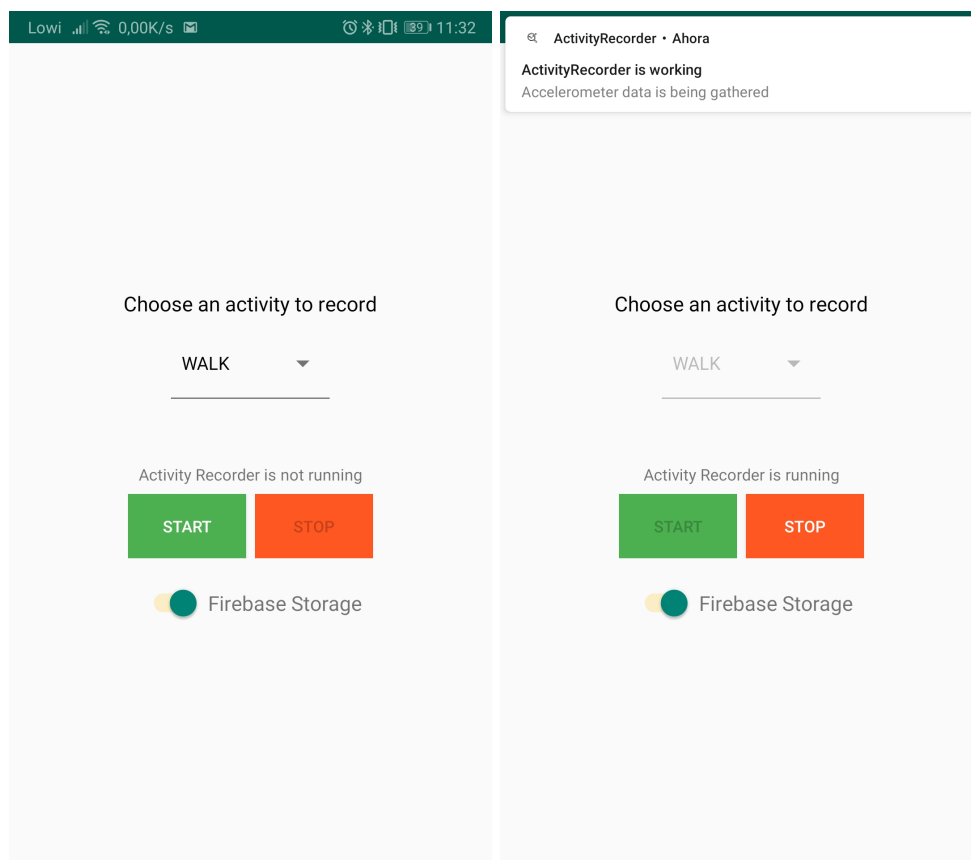


Figura 4.1: Arquitectura y flujo de trabajo de la aplicación ActivityRecorder.



(a) Inicio de aplicación

(b) En ejecución

Figura 4.2: Interfaz de la aplicación ActivityRecorder.

propio. Para evitar esto, cuando el usuario presiona el botón “STOP”, la aplicación elimina una determinada cantidad de registros iniciales y finales antes de guardarlos local y remotamente. La cantidad de registros eliminados se ha establecido en un valor de 250 (aproximadamente 5 segundos), tanto iniciales como finales.

El código correspondiente a este componente se encuentra disponible en el repositorio de GitHub **ActivityRecorder**².

4.2. Model Training

Google Cloud Platform ofrece diversos servicios y productos de computación en la nube que permiten desarrollar sistemas complejos con un coste reducido. Los principales servicios empleados para el desarrollo del sistema son:

- *Cloud Storage*³: permite el almacenamiento de objetos en la nube. Mediante este servicio se almacenan los registros de acelerómetro provenientes de la aplicación **ActivityRecorder** y otra información.
- *Cloud Functions*⁴: funciones como servicio (FaaS) que permiten la ejecución de pequeños fragmentos de código en la nube, sin necesidad de gestionar servidores. La ejecución de estas funciones puede estar dirigida por una gran variedad de eventos, como peticiones HTTP, acciones de creación de objetos en una instancia de *Cloud Storage* o mediante mensajes (*Cloud Pub/Sub*). Estas funciones se emplean en el sistema para tratamiento de datos y entrenamiento del modelo de predicción propio.
- *Cloud Pub/Sub*⁵: servicio de mensajería y transferencia de datos. Este servicio está compuesto por “tópicos”, los cuales sirven para publicar mensajes. Cuando otros servicios desean recibir los mensajes de un “tópico” solo tienen que suscribirse. Por otro lado, un servicio puede publicar un mensaje en cualquier “tópico”. Este servicio de mensajería se emplea en el sistema para comunicar entre sí ciertas *cloud functions*, las cuales se suscriben a un determinado “tópico” y se ejecutan cuando se publica un mensaje con dicho “tópico”.
- *Firebase Custom ML*⁶: permite la publicación de modelos de aprendizaje automático para su posterior descarga en dispositivos móviles.

En la Figura 4.3 se muestra el flujo de trabajo de la arquitectura *cloud* desarrollada. El sistema *cloud* empieza a funcionar al recibir los registros del acelerómetro provenientes de **ActivityRecorder** en *Cloud Storage*. Este hecho produce la ejecución de la *cloud function* `handle_storage_uploads`, la cual determina que se han subido nuevos registros de acelerómetro y publica un mensaje en el “tópico” `new_data` con la ruta del nuevo fichero de registros. A causa de publicar este mensaje, se lanza la ejecución de la *cloud function* `perform_feature_extraction`, la cual se encarga de extraer características (ver Sección 4.2.1) de los datos en crudo del acelerómetro y almacenarlos en *Cloud Storage*. Esta operación produce de nuevo la ejecución de

²<https://github.com/matey97/ActivityRecorder>

³Cloud Storage | Google Cloud

⁴Cloud Functions | Google Cloud

⁵Cloud Pub/Sub | Google Cloud

⁶Modelos personalizados | Firebase

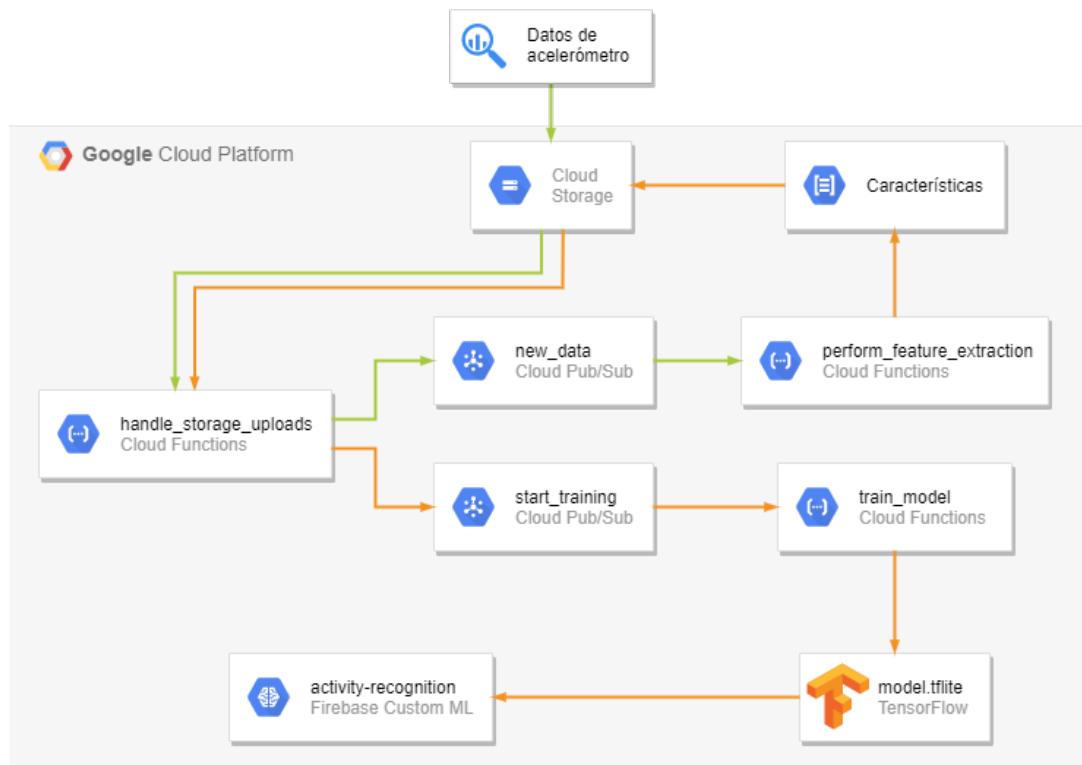


Figura 4.3: Flujo de trabajo *cloud*.

`handle_storage_uploads`, la cual determina que se han almacenado nuevas características y publica un mensaje en el “tópico” `start_training`. Dicho mensaje provoca la ejecución de la *cloud function* `train_model`, la cual se encarga de entrenar el modelo de reconocimiento propio con las nuevas características (ver Sección 4.2.2) y de publicarlo con formato **TensorFlow Lite**⁷ en *Firebase Custom ML* (requisito RF7).

La implementación de las tres *cloud functions* desarrolladas se encuentra disponible en el repositorio de GitHub **activity-detection-cloud-functions**⁸.

4.2.1. Extracción de características

En algunos casos, antes de realizar la extracción de características es necesario realizar algún tipo de preprocesado sobre la señal. Esto es lo que estudian Politi y otros [28], donde comparan los resultados de aplicar técnicas de preprocesado, como, por ejemplo, extraer la aceleración gravitacional o la aceleración del cuerpo sobre los datos en crudo, con los resultados de no aplicar ninguna técnica de preprocesado. Dichos experimentos revelan como los mejores resultados se obtienen sin aplicar ningún tipo de preprocesado a la señal antes de extraer las características. Por lo tanto, no se aplica ninguna técnica de preprocesado a los registros del acelerómetro antes de extraer las características.

En cuanto a la extracción de características, existen una gran cantidad de técnicas que pueden aplicarse en las señales de un sensor, en este caso, del acelerómetro [15]. Estas técnicas pueden dividirse, según su dominio de aplicación, en técnicas del dominio temporal, técnicas

⁷TensorFlow Lite | AA para dispositivos móviles

⁸<https://github.com/matey97/activity-detection-cloud-functions>

del dominio de la frecuencia y técnicas del dominio discreto. Por un lado, las técnicas del dominio temporal se basan en aplicar funciones estadísticas y matemáticas sobre subconjuntos de la señal, cuyo coste tanto temporal como espacial suele ser reducido. Por contra, en las técnicas del dominio de la frecuencia suele realizarse una transformada rápida de Fourier⁹ y aplicar posteriores operaciones sobre las componentes de la transformada, las cuales pueden llegar a ser bastante costosas. En cuanto a las técnicas del dominio discreto, éstas se basan en comparar dos señales mediante algoritmos para medir la similitud entre las mismas, como, por ejemplo, *Dynamic Time Warping* [33]. El coste de estas técnicas suele ser moderado, pero mayor que las del dominio temporal. Es por ello, que las técnicas empleadas para la extracción de características en este trabajo son las de dominio temporal.

Dentro de las técnicas del dominio temporal, las técnicas de extracción seleccionadas son algunas de las más empleadas en diversos artículos como [12], [20] o [3]:

- *Pitch* y *Roll*: rotación sobre los ejes X e Y del dispositivo móvil (ver Figura 4.4). Pese a que estas características suelen no utilizarse, emplearlas puede mejorar ligeramente la precisión de las predicciones [12].
- Funciones matemáticas y estadísticas: la aplicación de ciertas funciones matemáticas y estadísticas permite extraer características que facilitan la diferenciación entre actividades dinámicas y estáticas, posturas y miden la estabilidad de la señal [15]. Las funciones empleadas son: media, mediana, máximo, mínimo, desviación estándar, RMS (*root mean square*) y rango (diferencia entre el máximo y el mínimo).
- Otras funciones:
 - Diferencias entre muestras: calcula la diferencia media entre dos valores consecutivos de la señal. Esta técnica se ha empleado, para determinar el tipo de movimiento (de baja o alta intensidad) de los usuarios [34].
 - *Zero-crossings*: determina el número de veces que la señal cruza un determinado valor, el cual suele ser la media de la señal. Esta técnica ha sido empleada para detectar pasos y diferenciar entre caminar y correr [21].

De estas 11 técnicas de extracción de características, 9 de ellas se aplican sobre cada uno de los componentes de la señal (X, Y, Z) produciendo una característica por componente. Las otras dos restantes (*pitch* y *roll*) producen una única característica cada una. En total, se generan 29 características.

Otro detalle a destacar, es la forma en la que se produce la extracción de las características. Esta extracción no se realiza sobre toda la señal a la vez, si no que se aplica sobre subconjuntos superpuestos de la señal, una técnica llamada *sliding window with overlapping* (Ventana deslizante con superposición) empleada en gran variedad de artículos [7][3][29]. El tamaño de la ventana empleada es de un segundo (50 muestras) con una superposición del 50%, tal y como puede verse en la Figura 4.5.

4.2.2. Arquitectura y entrenamiento del modelo

Si bien numerosos artículos emplean redes neuronales para el reconocimiento de actividades, ya sea para la creación de nuevos sistemas [4] o para la comparación de métodos de clasifica-

⁹Transformada Rápida de Fourier | Wikipedia

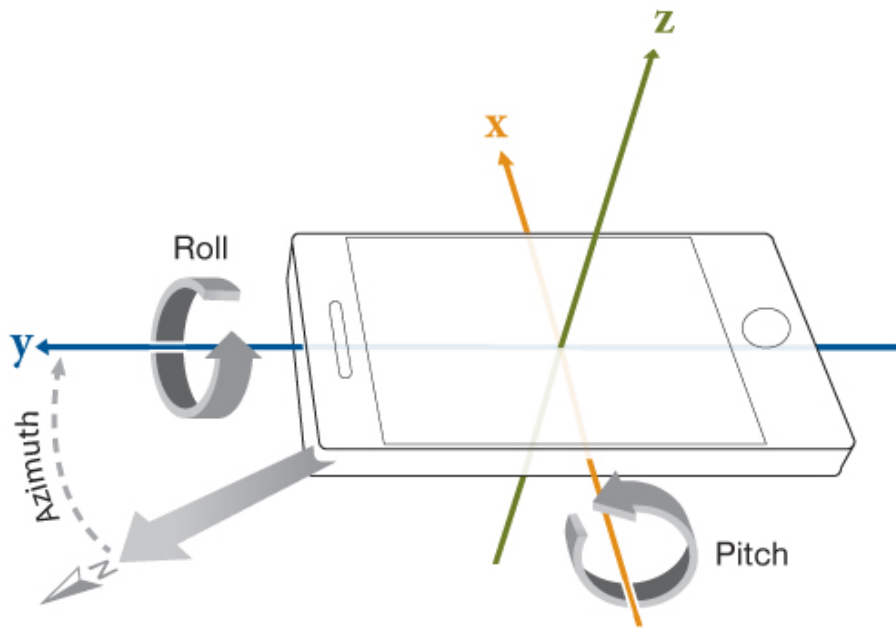


Figura 4.4: Esquema del *pitch* y el *roll* en un dispositivo móvil.

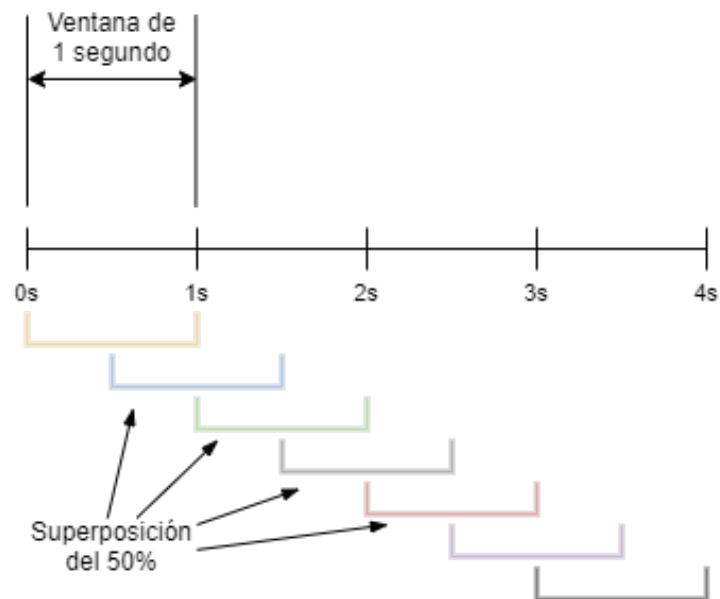


Figura 4.5: Esquema de ventana deslizante con superposición

ción [20][3][36], los autores no especifican la arquitectura de dichas redes. Por tanto, el modelo propuesto para el sistema de reconocimiento de actividades propio es una red neuronal con la siguiente arquitectura:

- Unidades de entrada: 29, una por cada característica extraída.
- Capas ocultas: una capa oculta de 512 unidades lineales rectificadas (ReLU¹⁰).
- Unidades de salida: cinco unidades con función de activación *SoftMax*¹¹, una por cada actividad especificada en la Sección 4.1.

Se ha seleccionado esta arquitectura, en concreto el número de capas ocultas y la cantidad de unidades por capa, debido a sus buenos resultados tanto en términos de precisión como de coste de entrenamiento, como se verá en la Sección 5.1.

Para el entrenamiento de la red propuesta se ha empleado la siguiente configuración:

- Algoritmo de optimización de parámetros: SGD (*Stochastic Gradient Descent*¹²) con las siguientes características:
 - *Learning rate*: 0.01, determina el grado en el que se actualizan los parámetros de la red, y por tanto, la velocidad de convergencia de la red.
 - *Decay*: 1e-6, factor de reducción del *learning rate* en cada ciclo de aprendizaje.
 - *Momentum*: 0.9, acelera el descenso por gradiente en la dirección correcta.
- Función de pérdida: *categorical crossentropy*, calcula la pérdida de entropía cruzada entre las predicciones y el resultado esperado. El objetivo del entrenamiento es maximizar esta función.
- *Batch size*: 50, tamaño de los grupos en los que se dividen los datos de entrenamiento, entrenando la red y actualizando los parámetros de la red tras emplear cada grupo.
- Épocas: 10, donde cada época supone un ciclo de aprendizaje sobre todas las muestras de entrenamiento.

4.3. Activity Recognition

4.3.1. Context APIs

El núcleo del sistema de detección de actividades sensible al contexto es el *plugin Context APIs*, el cual se encarga de estimar las actividades del usuario y de proporcionar su localización GPS. Este componente se ha implementado empleando el *framework* de desarrollo multiplataforma NativeScript ya que es la herramienta de desarrollo empleada actualmente en el proyecto SyMptOMS. El motivo de haber desarrollado estas funcionalidades como un *plugin* reside en

¹⁰ *Rectified Linear Unit* | Wikipedia

¹¹ *SoftMax function* | Wikipedia

¹² *Stochastic Gradient Descent* | Wikipedia

la posibilidad de que éste pueda ser empleado por desarrolladores externos al proyecto. Cabe destacar que, aunque el *framework* es multiplataforma (Android e iOS), su funcionalidad solo está implementada para dispositivos Android.

En la Figura 4.6 se muestra la arquitectura y el flujo de trabajo del *plugin*, el cual presenta las siguientes funcionalidades:

- **Reconocimiento de actividades:** permite escuchar cambios en las actividades del usuario empleando cualquiera de los tres sistemas de reconocimiento. El funcionamiento de dichos sistemas es el siguiente:
 - **ActivityTransition:** cuando se produce una transición entre actividades que realiza un usuario, el sistema notifica al usuario del *plugin* que actividad ha terminado y que otra actividad ha empezado a hacer el usuario (requisito RF2). Las actividades soportadas por este servicio son: circular en vehículo o en bicicleta, correr, caminar y estar parado. El sistema es efectivo en cuanto a precisión, pero no tanto en cuanto al retraso en las notificaciones respecto al momento en el que se realiza el cambio de actividad.
 - **ActivityUpdate:** cuando el sistema detecta un cambio significativo en la actividad que realiza el usuario, se notifica al usuario del *plugin* la actividad detectada (requisito RF3). Este sistema detecta, además de las actividades del sistema anterior, la variación de inclinación del dispositivo. El sistema también notifica cuando no le ha sido posible identificar la actividad. La precisión de este sistema no siempre es correcta, pero presenta un retardo aceptable entre el cambio de actividad y la notificación.
 - **Custom Recognizer:** este sistema hace uso de un modelo de red neuronal con la arquitectura descrita en la Sección 4.2.2 y formato **TensorFlow Lite**. Este modelo debe ser inyectado al *plugin* por el usuario del mismo, aportando flexibilidad a la hora de emplear distintos modelos, siempre que tengan la arquitectura descrita. El sistema se encarga de recoger registros del acelerómetro de forma similar a como lo hace la aplicación Android de la Sección 4.1, con la misma frecuencia de recogida de 50Hz. Cuando el sistema ha recogido 50 muestras del acelerómetro (aproximadamente 1 segundo), se procede a realizar la extracción de características especificadas en la Sección 4.2.1. Una vez extraídas las características, se introducen en la red neuronal y se obtiene la actividad estimada (requisito RF4). Como se ha dicho anteriormente, las actividades reconocibles por el modelo son: correr, caminar, estar quieto, estar sentado y estar sentado en cuclillas (estado de miedo). Cabe destacar, que al igual que en el proceso de extracción de características de la Sección 4.2.1, en el proceso de predicción se emplea la técnica de ventana deslizante con superposición. De esta forma, este sistema proporciona estimaciones de actividades cada 0,5 segundos.
- **Geolocalización:** permite obtener la localización del usuario cada un determinado periodo de tiempo configurable (aproximadamente). Para lograr esto, el *plugin* hace uso del API de **Android LocationManager**¹³, sobre la cual se pueden solicitar actualizaciones de ubicación GPS (requisito RF1).

El código correspondiente a este *plugin* puede encontrarse en el repositorio de GitHub **nativescript-context-apis**¹⁴.

¹³LocationManager | Android Developers

¹⁴<https://github.com/GeoTecINIT/nativescript-context-apis/tree/high-res>

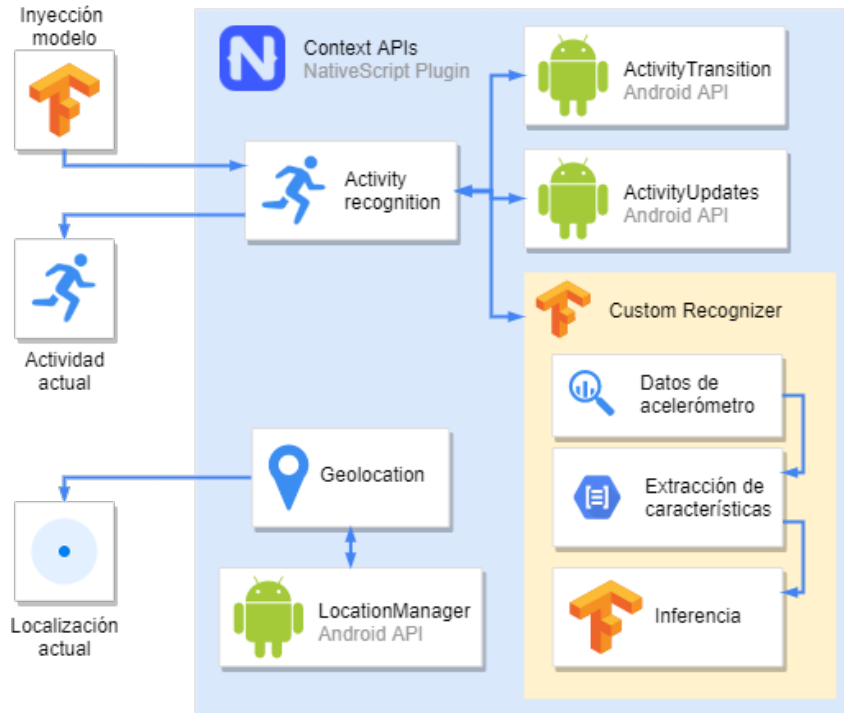


Figura 4.6: Arquitectura y flujo de trabajo del *plugin* Context APIs.

4.3.2. Context Activity Recognition

El último de los componentes que forman el sistema es la aplicación **Context Activity Recognition**, desarrollada, al igual que el *plugin* de la sección anterior, mediante el *framework* NativeScript. Esta es una aplicación demostradora para un caso de uso concreto que implementa el sistema de detección de actividades sensible al contexto propuesto, haciendo uso del *plugin* Context APIs.

En la Figura 4.7 se muestra la arquitectura y el flujo de trabajo de esta aplicación y como interactúa con los sistemas desarrollados en las Secciones 4.2 y 4.3.1. Como puede verse, la aplicación descarga el modelo entrenado de reconocimiento propio desde *Firestore Custom ML* y lo inyecta al *plugin* Context APIs. Además, la aplicación cuenta con otros dos componentes:

- **Geofencing:** determina, a partir de las localizaciones proporcionadas por el *plugin*, si el paciente se encuentra en un área de interés para su tratamiento o no, notificando de ello al **RecognitionManager**.
- **RecognitionManager:** se encarga de solicitar al *plugin* la activación o desactivación de sus sistemas de reconocimiento en función del contexto del paciente, teniendo en cuenta tanto su ubicación como su actividad previa.

En concreto, la aplicación se ha desarrollado para un caso de uso concreto, el cual se correspondería con la monitorización de un paciente con agorafobia. En este caso propuesto, un lugar de interés para el tratamiento del paciente sería un lugar abierto, como, por ejemplo, un parque. Por tanto, la aplicación establece una zona de **geofencing** sobre un determinado parque. Cuando el paciente se encuentre fuera de dicha zona, no se requerirá una monitorización estricta, por lo que se empleará el sistema **ActivityTransition**. Por otro lado, cuando el paciente entre en

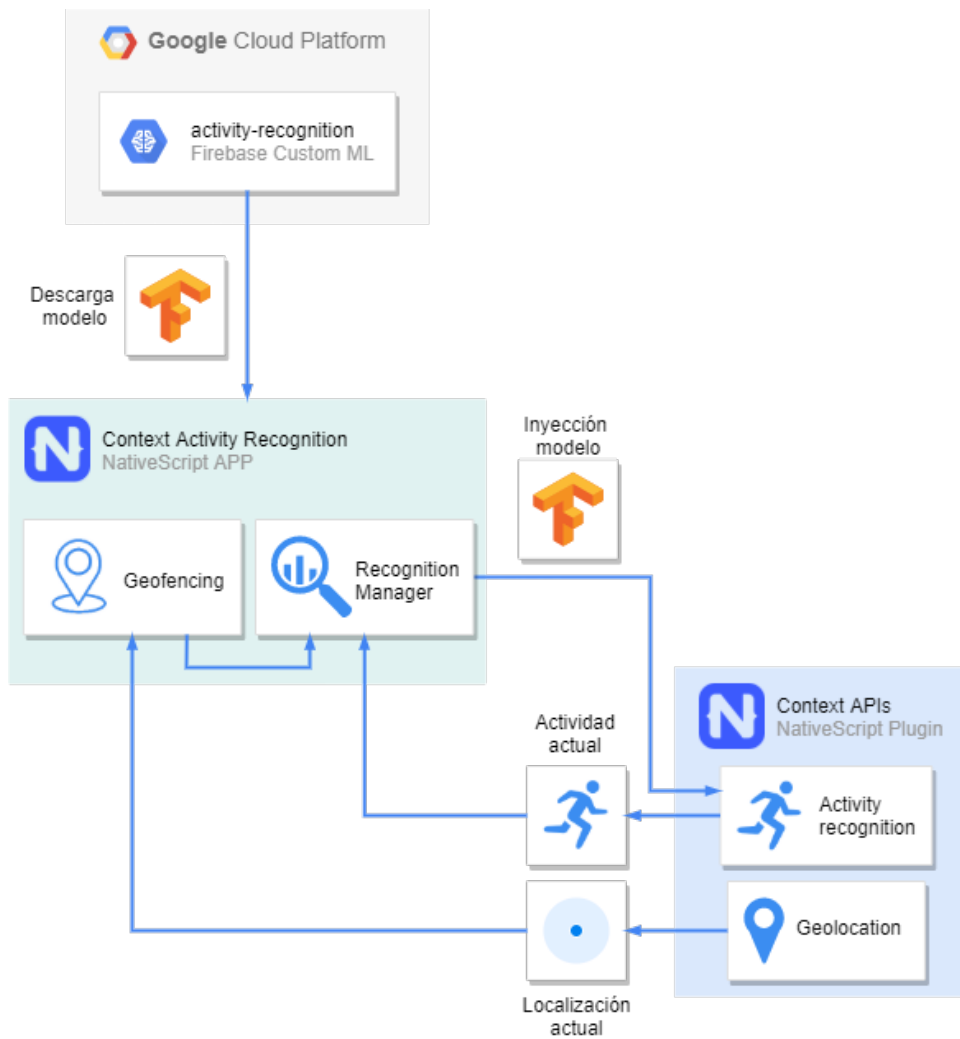


Figura 4.7: Arquitectura y flujo de trabajo de la aplicación Context Activity Recognition.

dicha área (por ejemplo, para una exposición terapéutica), se requerirá una monitorización con una mayor frecuencia, y, por tanto, se activará el sistema `ActivityUpdate`. Cuando con este sistema se detecte una determinada actividad, en este caso, que el paciente se quede parado, se activará el sistema `CustomRecognizer` para tener una monitorización continua y estricta. En la Figura 4.8 se muestra un diagrama de flujo de la funcionalidad descrita, con la cual se cumpliría el requisito RF5 definido en la Sección 3.1.

Aunque queda fuera del alcance de este proyecto, esta información proveniente de la monitorización sería analizada posteriormente por el terapeuta del paciente, o incluso podría ser empleada para realizar alguna acción en tiempo real mientras el paciente se somete a la exposición, es decir, como parte de una intervención terapéutica.

El código fuente de esta aplicación puede encontrarse en el repositorio de GitHub **context-activity-recognition**¹⁵

¹⁵<https://github.com/matey97/context-activity-recognition>

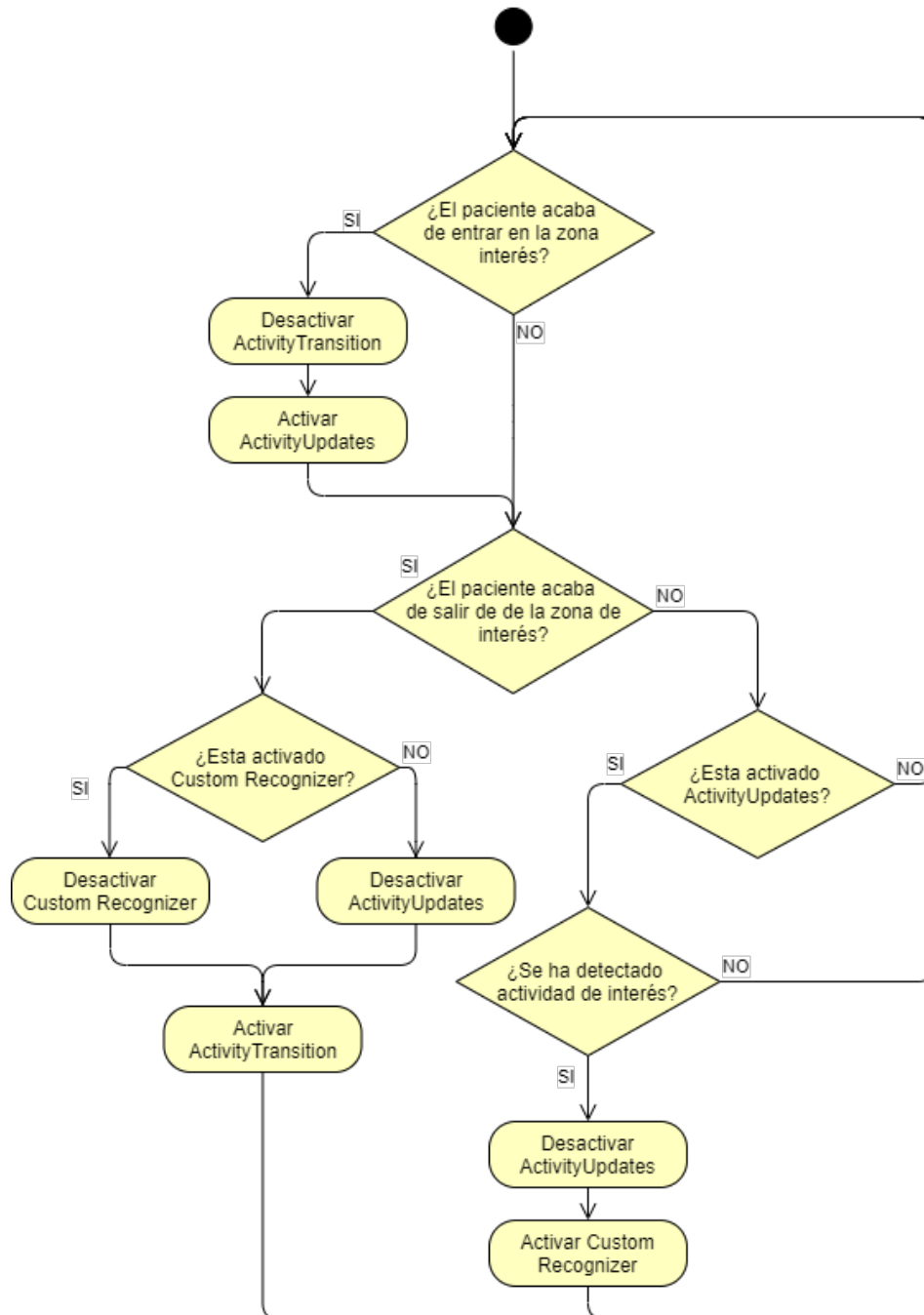


Figura 4.8: Diagrama de flujo del caso de uso de la aplicación Context Activity Recognition.

Capítulo 5

Evaluación

En este capítulo se muestra la evaluación del rendimiento del sistema de reconocimiento de actividades propio y del funcionamiento general del sistema de detección de actividades sensible al contexto desarrollado.

5.1. Evaluación del modelo de reconocimiento propio

El objetivo de esta sección es determinar si el rendimiento del modelo propuesto, en términos de precisión, presenta unos resultados adecuados, y por tanto, que cumple el requisito RR1.

El modelo de red neuronal propuesto y descrito en la Sección 4.2.2 ha sido entrenado para su evaluación con un total de 8486 muestras de las distintas actividades propuestas: correr (R), estar sentado (S), estar sentado en cuclillas (SF, actividad de interés psicológico), estar quieto/de pie (ST) y caminar (W). En la Tabla 5.1 se muestra la cantidad de muestras para cada una de dichas actividades. La precisión del modelo ha sido evaluada mediante 5909 muestras de prueba de las actividades propuestas. La cantidad de muestras correspondiente a cada actividad se muestra en la Tabla 5.2. Las muestras han sido recogidas depositando el dispositivo móvil en el bolsillo izquierdo del pantalón, tanto hacia arriba como hacia abajo.

Actividad	Cantidad
R	1543
S	1735
SF	1723
ST	1699
W	1786

Tabla 5.1: Muestras de entrenamiento por actividad.

En la Tabla 5.3 se muestra la matriz de confusión con los resultados de la evaluación del modelo con los datos de prueba. Como puede verse, el modelo muestra una gran precisión a la hora de predecir las actividades R, S y SF. Por otro lado, la precisión del modelo es ligeramente menor para la actividad W, la cual se confunde en ocasiones con la actividad ST. Del mismo modo, la actividad ST presenta una precisión menor, siendo confundida con la actividad SF. Esto puede ser debido a que algunos movimientos puntuales (ej. inclinación) durante la recogida

Actividad	Cantidad
R	1184
S	1234
SF	1180
ST	1166
W	1145

Tabla 5.2: Muestras de prueba por actividad.

		Predicciones				
		R	S	SF	ST	W
Originales	R	1182	0	0	1	1
	S	0	1232	0	2	0
	SF	0	0	1180	0	0
	ST	0	0	141	1025	0
	W	0	0	0	48	1097

Tabla 5.3: Matriz de confusión.

de muestras para la actividad ST pueden ser confundidos con la actividad SF.

En la Tabla 5.4 se muestran las métricas de precisión, sensibilidad y F_1 score¹ (ponderación entre precisión y sensibilidad) de cada una de las actividades extraídas de la matriz de confusión. Como puede verse, el rendimiento del sistema es generalmente bueno en todas las actividades propuestas, siendo lo más problemático la distinción entre las actividades SF y ST, y entre ST y W.

	Precisión	Sensibilidad	F_1 score
R	1	0,998	0,999
S	1	0,998	0,999
SF	0,893	1	0,943
ST	0,952	0,879	0,914
W	0,999	0,958	0,978
Media	0,968	0,968	0,966

Tabla 5.4: Precisión, sensibilidad y F_1 score del sistema con los datos de prueba.

Por último, se ha querido realizar la evaluación del modelo de reconocimiento una vez desplegado en el dispositivo móvil, probándolo con datos inmediatamente recogidos por el dispositivo durante un experimento. En dicho experimento, el usuario ha estado quieto (ST) durante un pequeño periodo de tiempo, tras el cual ha empezado a caminar (W). A continuación, el usuario se ha sentado (S) durante un momento antes de sentarse en cuclillas (ST). Posteriormente, el usuario se ha vuelto a sentar (S) y, finalmente, se ha levantado y se ha puesto a caminar (W). En la Figura 5.1 pueden verse las predicciones que ha realizado el modelo durante el transcurso experimento descrito, donde se puede destacar lo siguiente:

- Durante la primera parte del experimento, el modelo predice sin errores las actividades ST y W.
- El modelo predice durante un instante que el usuario está corriendo, aunque este hecho

¹*F1 score* - Wikipedia

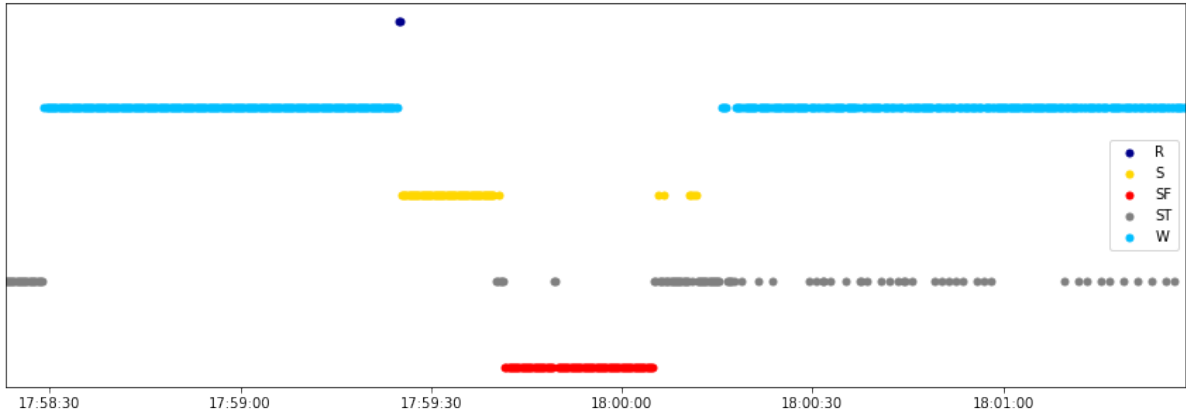


Figura 5.1: Representación de actividades predichas por el modelo.

es debido a la transición que se produce entre las actividades W y S.

- Durante la transición entre S y SF, y puntualmente unos momentos después, el modelo predice la actividad ST. Esto concuerda con la problemática en la distinción entre las actividades ST y SF comentado anteriormente.
- En la transición entre SF y S, el modelo predice de forma incorrecta la actividad ST durante un buen periodo de tiempo, algo que no se podía ver con los datos de prueba anteriores.
- En la parte final del experimento, tras la transición de S a W, el modelo predice correctamente la actividad gran parte del tiempo. Por otro lado, puede verse claramente como el modelo produce falsos positivos de la actividad ST, lo cual concuerda con la dificultad en la distinción entre las actividades ST y W vista anteriormente. En este caso, al estudiar la confianza de las predicciones del modelo, se observa con la confianza media de las actividades W es de 0,959, mientras que la confianza de las actividades ST es de 0,749. Teniendo en cuenta estos valores, las predicciones del modelo podrían mejorarse con post-procesamiento, teniendo en cuenta las últimas predicciones y sus respectivas probabilidades.

En las Tablas 5.5 y 5.6 se muestran la matriz de confusión y las métricas de precisión, sensibilidad y F_1 score en el experimento con el sistema desplegado. En estas tablas no se incluye la actividad R ya que no se realizó durante el experimento. Debe destacarse cómo la precisión para la actividad ST es bastante reducida debido a la gran cantidad de falsos positivos. En general, y tratándose de resultados con el modelo desplegado en el dispositivo móvil, los resultados son positivos, aunque mejorables.

		Predicciones			
		S	SF	ST	W
Originales	S	66	0	20	0
	SF	0	92	7	0
	ST	0	0	38	0
	W	0	0	43	476

Tabla 5.5: Matriz de confusión con el sistema desplegado.

Debido a los resultados obtenidos tanto en la evaluación mediante datos de prueba como

	Precisión	Sensibilidad	F_1 score
S	1	0,767	0,868
SF	1	0.929	0,963
ST	0,351	1	0,519
W	1	0,917	0,956
Media	0,837	0,875	0,826

Tabla 5.6: Precisión, sensibilidad y F_1 score del sistema una vez desplegado.

con el experimento realizado, puede decirse que se ha cumplido con el objetivo O1 y el requisito de rendimiento RR1.

5.2. Evaluación de la selección de sistema de reconocimiento dependiendo del contexto

En esta sección se busca evaluar el funcionamiento del sistema de detección de actividades sensible al contexto propuesto. Para esta evaluación se ha realizado un experimento simulando el caso de uso especificado en secciones anteriores. En concreto, en este experimento se simula la exposición a un espacio abierto de un paciente de agorafobia.

Para ello, se ha empleado la aplicación demostradora **Context Activity Recognition** especificada en la Sección 4.3.2. En dicha aplicación, se ha configurado como lugar de interés un parque, sobre el cual se establece una *geofence*. Además, se registran en un fichero de texto todas las actividades reconocidas por los diferentes sistemas y los eventos de entrada y salida de la *geofence* para llevar a cabo la evaluación del sistema.

En la Figura 5.2 se representan las actividades registradas por el sistema de detección en función del contexto. Los cuadrados azules se corresponden con las transiciones de actividades reportadas por el sistema **ActivityTransition**, el cual es empleado fuera de la zona de interés. La entrada a dicha zona de interés viene representada por la línea vertical verde, y puede verse como tras la entrada del paciente en dicha área, el sistema **ActivityUpdates** comienza a reportar actividades (representadas por estrellas amarillas). En el momento en que se detecta una actividad de interés, representada por la línea vertical amarilla, se empieza a emplear el sistema de reconocimiento propio, el cual reporta actividades a una mayor frecuencia. Tras abandonar el área de interés (línea vertical roja), el sistema de reconocimiento propio deja de proporcionar predicciones y se empieza a emplear el sistema **ActivityTransition**.

Un detalle a destacar es el retardo que puede apreciarse entre que se detecta la actividad de interés y que el reconocedor propio empieza a proporcionar predicciones. Además, puede verse como tras detectar la actividad de interés, se han detectado otras actividades posteriormente, con lo que puede parecer que existe un determinado retardo entre la desactivación y activación de estos sistemas. Tras analizar en detalle los datos, se ha determinado que, efectivamente, existe un cierto retardo, pero no dentro del sistema desarrollado, sino dentro del sistema **ActivityUpdates**. En este análisis se ha observado que el sistema ha proporcionado actualizaciones de actividades de forma desordenada. Es decir, dentro del fichero de texto donde se guardan los registros, existen unas pocas actividades con fecha anterior a otras actividades notificadas con anterioridad y con fecha posterior. Este hecho lleva a la hipótesis de que dicho sistema puede “retractarse” de sus predicciones anteriores con nueva información, proporcio-

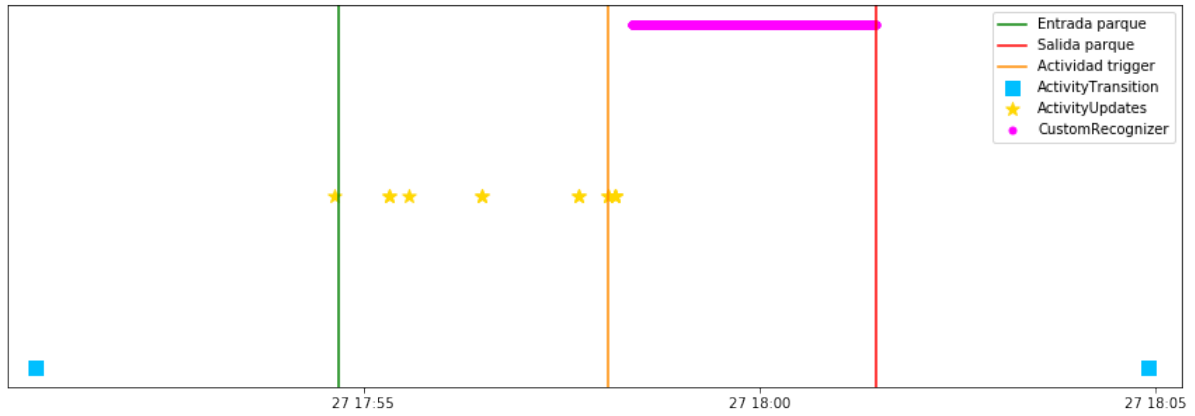


Figura 5.2: Representación de actividades predichas por el sistema en función del contexto.

nando nuevas predicciones pero con fecha anterior a actualizaciones previas. Por tanto, dicho retraso que puede apreciarse en la figura, no sería real.

Por los resultados obtenidos en el experimento de evaluación del sistema de detección sensible al contexto, puede decirse que se ha cumplido el objetivo O2, así como los requisitos funcionales relacionados (RF1–RF5). Por otra parte, y con respecto al requisito RR2, pese a que no se ha validado mediante pruebas específicas, no se aprecia un sobre coste por el uso del sistema de reconocimiento propio en el sistema atendiendo a los datos recogidos durante el experimento.

Capítulo 6

Conclusiones

En este trabajo se ha presentado un sistema de detección de actividades sensible al contexto, cuyo objetivo es seleccionar un sistema de reconocimiento de actividades u otro dependiendo de la situación del usuario. Este sistema permite emplear los dos servicios que proporciona Android para el reconocimiento de actividades, además de un sistema de reconocimiento propio basado en una red neuronal entrenada con unas actividades específicas. El trabajo desarrollado incluye: una aplicación Android que permite la recogida de muestras del acelerómetro para el entrenamiento de dichas actividades; una arquitectura *cloud* para el tratamiento de estas muestras, el entrenamiento automático de la red neuronal de reconocimiento de actividades y la publicación de dicho modelo; un *plugin* para el *framework* de desarrollo NativeScript, el cual permite la configuración y la utilización de los sistemas de Android y del modelo propio para el reconocimiento de actividades, y una aplicación demostradora con un caso de uso que emplea los sistemas de reconocimiento de actividades del *plugin* anterior dependiendo del contexto del usuario. También, se ha evaluado la precisión de la red neuronal entrenada, obteniendo buenos resultados con datos de entrenamiento y unos resultados un poco peores con el modelo directamente desplegado en el dispositivo, aunque existe margen de mejora refinando y post-procesando las estimaciones de la red. Finalmente, se ha comprobado el correcto funcionamiento del sistema de detección sensible al contexto, el cual ha empleado los sistemas de reconocimiento disponibles dependiendo del contexto del usuario. Este sistema desarrollado se integrará en el proyecto SyMptOMS donde será de utilidad para la monitorización y seguimiento de pacientes con diversos trastornos de salud mental.

Desde un punto de vista técnico, ambos objetivos planteados en este trabajo se han cumplido de forma satisfactoria, tanto el desarrollo del modelo propio de detección de actividades (O1) como del sistema de detección de actividades sensible al contexto (O2).

Para el futuro, se han identificado ciertas mejoras que pueden implementarse en el sistema desarrollado y que pueden ser de utilidad. Algunas de estas mejoras son:

- Incluir un modelo por defecto en **Context APIs**. Esto permitiría que desarrolladores que no contasen con un modelo para inyectar en el *plugin* pudiesen emplear el sistema de reconocimiento basado en la red neuronal.
- Permitir añadir nuevas actividades para su detección desde la aplicación **ActivityRecorder**. Esto añadiría una gran flexibilidad al sistema pero conllevaría cambios menores en dicha aplicación y en *plugin Context APIs*, y cambios más sofisticados en la arquitectura *cloud*.

- Permitir la configuración de la frecuencia de predicción del sistema de reconocimiento propio. Actualmente, el sistema de reconocimiento propio realiza predicciones cada 0,5 segundos, donde esta frecuencia depende del tamaño de la *slidding window*. Permitir variar esta frecuencia añadiría flexibilidad al sistema, aunque requeriría de más modelos, ya que estos se entrenan con características extraídas con un tamaño de *slidding window* determinado.

Por último, y personalmente, este trabajo ha supuesto un reto a nivel de desarrollo por la mezcla de varias tecnologías y por las dudas surgidas, en ocasiones, de si el sistema llegaría a funcionar. En general, estoy muy satisfecho con el trabajo realizado y espero poder desarrollar las mejoras indicadas anteriormente e integrar el sistema desarrollado dentro del proyecto SyMptOMS.

Bibliografía

- [1] Jonghun Baik, Geehyuk Lee, Wonbae Park, and Byoung-Ju Yun. Accelerometer signal processing for user activity detection. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 610–617. Springer, 2004.
- [2] Ling Bao and Stephen S Intille. Activity recognition from user-annotated acceleration data. In *International conference on pervasive computing*, pages 1–17. Springer, 2004.
- [3] Akram Bayat, Marc Pomplun, and Duc A Tran. A study on human activity recognition using accelerometer data from smartphones. *Procedia Computer Science*, 34:450–457, 2014.
- [4] Serge Thomas Mickala Bourobou and Younghwan Yoo. User activity recognition in smart homes using pattern clustering applied to temporal ann algorithm. *Sensors*, 15(5):11953–11971, 2015.
- [5] Nathalie Bricon-Souf and Conrad R Newman. Context awareness in health care: A review. *international journal of medical informatics*, 76(1):2–12, 2007.
- [6] Liang Cao, Yufeng Wang, Bo Zhang, Qun Jin, and Athanasios V Vasilakos. Gchar: An efficient group-based context—aware human activity recognition on smartphone. *Journal of Parallel and Distributed Computing*, 118:67–80, 2018.
- [7] Pierluigi Casale, Oriol Pujol, and Petia Radeva. Human activity recognition from accelerometer data using a wearable device. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 289–296. Springer, 2011.
- [8] Saisakul Chernbumroong, Anthony S Atkins, and Hongnian Yu. Activity classification using a single wrist-worn accelerometer. In *2011 5th International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA) Proceedings*, pages 1–6. IEEE, 2011.
- [9] Fang-Chen Chuang, Jeen-Shing Wang, Ya-Ting Yang, and Tzu-Ping Kao. A wearable activity sensor system and its physical activity classification scheme. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2012.
- [10] Wan-Young Chung, Amit Purwar, and Annapurna Sharma. Frequency domain approach for activity classification using accelerometer. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1120–1123. IEEE, 2008.
- [11] Maria Cornacchia, Koray Ozcan, Yu Zheng, and Senem Velipasalar. A survey on activity detection and classification using wearable sensors. *IEEE Sensors Journal*, 17(2):386–403, 2016.
- [12] Doruk Coskun, Ozlem Durmaz Incel, and Atay Ozgovde. Phone position/placement detection using accelerometer: Impact on activity recognition. In *2015 IEEE Tenth International*

Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pages 1–6. IEEE, 2015.

- [13] Davide Curone, Gian Mario Bertolotti, Andrea Cristiani, Emanuele Lindo Secco, and Giovanni Magenes. A real-time and self-calibrating algorithm based on triaxial accelerometer signals for the detection of human posture and activity. *IEEE transactions on information technology in biomedicine*, 14(4):1098–1105, 2010.
- [14] Anind K Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.
- [15] Davide Figo, Pedro C Diniz, Diogo R Ferreira, and Joao MP Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010.
- [16] Emma Fortune, Marie Tierney, Cliodhna Ni Scanail, Ala Bourke, Norelee Kennedy, and John Nelson. Activity level classification algorithm using shimmerTM wearable sensors for individuals with rheumatoid arthritis. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3059–3062. IEEE, 2011.
- [17] Gye Hwan Jin, Sang Bock Lee, and Tae Soo Lee. Context awareness of human motion states using accelerometer. *Journal of Medical Systems*, 32(2):93–100, 2008.
- [18] Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE transactions on information technology in biomedicine*, 10(1):156–167, 2006.
- [19] Asad Masood Khattak, Dang Viet Hung, Phan Tran Ho Truc, Le Xuan Hung, D Guan, Zeeshan Pervez, Manhyung Han, Sungyoung Lee, Young-Koo Lee, et al. Context-aware human activity recognition and decision making. In *The 12th IEEE International Conference on e-Health Networking, Applications and Services*, pages 112–118. IEEE, 2010.
- [20] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.
- [21] Seon-Woo Lee and Kenji Mase. Activity and location recognition using wearable sensors. *IEEE pervasive computing*, 1(3):24–32, 2002.
- [22] Andrea Mannini and Angelo Maria Sabatini. On-line classification of human activity and estimation of walk-run speed from acceleration data using support vector machines. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3302–3305. IEEE, 2011.
- [23] Jani Mantyjarvi, Johan Himberg, and Tapio Seppanen. Recognizing human motion with multiple acceleration sensors. In *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236)*, volume 2, pages 747–752. IEEE, 2001.
- [24] Jenny Margarito, Rim Helaoui, Anna M Bianchi, Francesco Sartor, and Alberto G Bonomi. User-independent recognition of sports activities from a single wrist-worn accelerometer: A template-matching-based approach. *IEEE Transactions on Biomedical Engineering*, 63(4):788–796, 2015.
- [25] David Naranjo-Hernández, Laura M Roa, Javier Reina-Tosina, and Miguel Angel Estudillo-Valderrama. Som: a smart sensor for human activity monitoring and assisted healthy ageing. *IEEE transactions on biomedical engineering*, 59(11):3177–3184, 2012.

- [26] Fajri Nurwanto, Igi Ardiyanto, and Sunu Wibirama. Light sport exercise detection based on smartwatch and smartphone using k-nearest neighbor and dynamic time warping algorithm. In *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 1–5. IEEE, 2016.
- [27] Arpit Parmar and Pragya Sharma. Ecological momentary interventions delivered by smartphone apps: Applications in substance use treatment in indian scenario. *Indian journal of psychological medicine*, 39(1):102, 2017.
- [28] Olga Politi, Iosif Mporas, and Vasileios Megalooikonomou. Comparative evaluation of feature extraction methods for human motion detection. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 146–154. Springer, 2014.
- [29] Olga Politi, Iosif Mporas, and Vasileios Megalooikonomou. Human motion detection in daily activity tasks using wearable sensors. In *2014 22nd European signal processing conference (EUSIPCO)*, pages 2315–2319. IEEE, 2014.
- [30] Si-Jung Ryu and Jong-Hwan Kim. Classification of long-term motions using a two-layered hidden markov model in a wearable sensor system. In *2011 IEEE International Conference on Robotics and Biomimetics*, pages 2975–2980. IEEE, 2011.
- [31] Sohrab Saeb, Emily G Lattie, Stephen M Schueller, Konrad P Kording, and David C Mohr. The relationship between mobile phone location sensor data and depressive symptom severity. *PeerJ*, 4:e2537, 2016.
- [32] Sohrab Saeb, Mi Zhang, Christopher J Karr, Stephen M Schueller, Marya E Corden, Konrad P Kording, and David C Mohr. Mobile phone sensor correlates of depressive symptom severity in daily-life behavior: an exploratory study. *Journal of medical Internet research*, 17(7):e175, 2015.
- [33] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.
- [34] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and Fei Lung Wong. Sensay: a context-aware mobile phone. In *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.*, pages 248–249, 2003.
- [35] Sa-kwang Song, Jaewon Jang, and Soojun Park. A phone for human activity recognition using triaxial acceleration sensor. In *2008 Digest of Technical Papers-International Conference on Consumer Electronics*, pages 1–2. IEEE, 2008.
- [36] AS Abdull Sukor, A Zakaria, and N Abdul Rahim. Activity recognition using accelerometer sensor and machine learning classifiers. In *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*, pages 233–238. IEEE, 2018.
- [37] Konlakorn Wongpatikaseree, Mitsuru Ikeda, Marut Buranarach, Thepchai Supnithi, Azman Osman Lim, and Yasuo Tan. Activity recognition using context-aware infrastructure ontology in smart home domain. In *2012 Seventh International Conference on Knowledge, Information and Creativity Support Systems*, pages 50–57. IEEE, 2012.
- [38] Chun Zhu and Weihua Sheng. Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(3):569–573, 2011.

- [39] Yingying Zhu, Nandita M Nayak, and Amit K Roy-Chowdhury. Context-aware activity recognition and anomaly detection in video. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):91–101, 2012.