



UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Trabajo Fin de Máster del
Máster Universitario en Ingeniería y Ciencia de Datos

**Análisis de pulsera inteligente para la detección de
estados afectivos mediante aprendizaje no supervisado
sobre series temporales**

Carlos Ilia Herráiz Montalvo

Dirigido por: Olga C. Santos Martín

Raúl Cabestrero Alonso

Curso: 2021-2022: 1ª Convocatoria

Agradecimientos

Dedico este trabajo a Salvador por ser mi inspiración.

Resumen

La utilización de relojes y pulseras inteligentes es una tendencia que cada día está más en auge. La presente memoria busca estudiar las posibilidades de estos dispositivos comerciales para identificar cambios en el estado afectivo durante la realización de las tareas de resolución de problemas con el fin de evaluar su posible aplicabilidad al Marco Lógico Colaborativo. El proyecto aborda la construcción de una plataforma para la recogida de datos, el estudio comparativo de la aplicación a series temporales de diversos algoritmos de aprendizaje automático y la visualización de resultados en tiempo real.

Dentro de la tarea de extracción de los datos se implementará una aplicación específica para el reloj inteligente que permitirá construir a partir de los datos disponibles de los sensores, un conjunto de datos mediante el uso de una API. En el análisis de los datos se aplicarán diferentes algoritmos de agrupamiento a las series temporales. Se estudiará la aplicación de algoritmos a las series temporales mediante cambios en la representación de los datos, así como otros algoritmos que procesan las series temporales directamente.

Finalmente se realiza una implementación que permite realizar una actividad y en tiempo real extraer los datos, enviarlos a un algoritmo de agrupamiento y visualizar en todo momento la evolución de la experiencia. Cuando el algoritmo señala que hay una alteración, el resultado es mostrado en la pulsera, permitiendo al usuario recibir este dato como estímulo visual y táctil.

Abstract

The use of smart watches and bracelets is a trend that is growing every day. This report seeks to study the possibilities of these commercial devices to identify changes in the affective state during the performance of problem-solving tasks in order to evaluate its possible applicability to the Collaborative Logical Framework. The project addresses the construction of a platform for data collection, the comparative study of the application of various machine learning algorithms to time series and the visualization of results in real time.

Within the task of extracting the data, a specific application for the smart watch will be implemented that will allow the construction of a set of data from the data available from the sensors, through the use of an API. In the analysis of the data, different grouping algorithms will be applied to the time series. The application of algorithms to time series through changes in the representation of the data will be studied, as well as other algorithms that process time series directly.

Finally, an implementation is carried out that allows an activity to be carried out and in real time to extract the data, send it to a grouping algorithm and visualize the evolution of the experience at all times. When the algorithm indicates that there is an alteration, the result is displayed on the smart watch, allowing the user to receive this data as a visual and tactile stimulus.

Índice general

1. Introducción	1
1.1. Estado del arte	2
1.1.1. Selección de dispositivos	2
1.1.2. Señales	3
1.1.3. Etiquetación de los datos	3
1.1.4. Algoritmos de aprendizaje automático	4
1.2. Motivación	4
1.3. Objetivos	5
1.4. Fases del trabajo	6
1.5. Estructura del documento	7
2. Materiales y métodos	9
2.1. Obtención de datos	9
2.2. Exploración	13
2.3. Preparación	17
2.4. Prueba de modelos	19
2.4.1. K-Medias	19
2.4.2. Propagación por afinidad	20
2.4.3. Agrupamiento espectral	21
2.4.4. BIRCH	22
2.4.5. Agrupamiento por aglomeración	23
2.4.6. Mean-Shift	24
2.4.7. DBSCAN	25
2.4.8. K-Medias con DTW	25
2.4.9. K-Shape	28
2.5. Ajuste	28
2.5.1. Stacking	29
2.5.2. Adaptación con DTW	31
2.6. Despliegue	32

3. Resultados	35
4. Conclusiones	39

Nomenclatura

- API Application Programming Interface, página 10
- BIRCH Balanced Iterative Reducing and Clustering using Hierarchies, página 19
- CSV comma-separated values, página 13
- DTW Dynamic Time Warping, página 25
- EDA Electroderminal Activity, página 3
- FC Frecuencia Cardíaca, página 3
- GMS Servicios de Google para Móviles, página 11
- IBI Interbeat Interval, página 3
- JSON JavaScript Object Notation, página 13
- KNN K nearest neighbour, página 4
- LED Light Emitting Diode, página 9
- MLC Marco Lógico Colaborativo, página 5
- RF Ramdom Forest, página 4
- SBD Shape-based distance, página 28
- SVM Support Vector Machine, página 4
- VRC Variabilidad del Ritmo Cardíaco, página 3

Índice de figuras

2.1. Etapas del proyecto	10
2.2. Arquitectura de aplicación Fitbit	12
2.3. Interfaz aplicación del reloj	14
2.4. Serie temporales de registradas	16
2.5. Representación mediante indicadores	18
2.6. Pruebas del modelo K-Medias con FC	20
2.7. Pruebas del modelo de agrupamiento espectral con FC	22
2.8. Prueba del modelo de agrupamiento BIRCH con FC	23
2.9. Prueba del modelo de agrupamiento aglomerativo con FC	24
2.10. Prueba del modelo Mean-Shift con FC	25
2.11. Comparación entre distancia euclidea y DTW.	26
2.12. Prueba del modelo K-Medias con DTW con FC	27
2.13. Detalle de agrupamiento de K-Medias con FC	27
2.14. Prueba de modelo K-Shape con FC	28
2.15. Stacking en serie Rubik.	31
2.16. Esquema de la construcción y despliegue del modelo	33
2.17. Visualización en tiempo real de FC	34
3.1. Distribución de segmentos en grupos por algoritmo K-Medias	36
3.2. Distribución de segmentos en grupos por algoritmo de agrupamiento espectral	36
3.3. Coeficiente silueta en función del parámetro número de grupos en modelo K-Medias para la serie Rubik	37

Índice de tablas

2.1. Fragmento de serie temporal	15
2.2. Mejores número de grupos para la serie temporal Rubik.	30
2.3. Mejores número de grupos para la serie temporal Python.	30
2.4. Mejores número de grupos para la serie temporal Drama.	30

Capítulo 1

Introducción

Es muy probable que nosotros mismos o algunas de las personas que en este momento están a nuestro alrededor, usen un reloj inteligente. Estos dispositivos cargados de sensores en su interior generan una gran cantidad de datos que son procesados y mostrados a través de las aplicaciones desarrolladas para ellos. Ante esta potencial disponibilidad de datos surge la pregunta de si sería posible aplicar un modelo de aprendizaje automático alimentado por los datos del reloj inteligente mientras se realiza una actividad. La actividad consistiría en realizar una labor de aprendizaje o resolución de problemas con el fin de detectar mediante el algoritmo si hay alguna alteración afectiva en el usuario del reloj.

Un reloj inteligente es un dispositivo evolucionado del reloj de pulsera digital que permite al usuario utilizar diferentes tecnologías inalámbricas, ejecutar aplicaciones, así como guardar y sincronizar datos captados de los diversos sensores. No obstante, el reloj inteligente ha sido concebido para servir de interfaz de un sistema remoto, por lo tanto, se puede considerar a día de hoy una extensión del teléfono inteligente. La popularidad de este tipo de dispositivos aumenta paulatinamente debido a que incluyen capacidades para monitorizar el ejercicio físico y diferentes aspectos de salud. Se espera que en un futuro próximo el reloj inteligente contenga de manera independiente todas las funciones del teléfono inteligente.

El mercado global de relojes inteligentes en 2021 fue valorado con un volumen de 68,59 millones de unidades en 2020, y se espera alcanzar un total de 230,30 millones de unidades en 2026. El mercado de relojes inteligentes tuvo un valor de 22.460 millones de dólares en 2021, y la previsión es que supere los 97.520 millones de dólares en 2028 [16]. Estas cifras implican que el mercado de teléfonos inteligentes tenga una tasa de crecimiento anual compuesto estimada de 21,5%. Este aumento vertiginoso del uso de este tipo de dispositivos y el interés porque cada vez incorporen más capacidades abre la puerta a que se apliquen técnicas avanzadas de procesamiento de datos para enriquecer aún más la experiencia de usuario, así como para ampliar el ámbito de aplicación de este tipo de dispositivos.

El avance de la tecnología de procesamiento de datos ha permitido en la última década la eclosión

de ramas de la computación como pueden ser procesamiento de datos masivos, así como un mayor desarrollo dentro del ámbito de la inteligencia artificial de las técnicas de aprendizaje automático¹. También existe gran eclosión de aprendizaje profundo aupado entre otros motivos por las nuevas arquitecturas hardware para el tratamiento de la información. Uno de los principales aceleradores de la aplicación de estos ámbitos de la inteligencia artificial es el desarrollo de aplicaciones para el sector de la salud, crecimiento que se hace muy evidente a partir de la pandemia de COVID-19. Se espera una tasa de crecimiento anual compuesto de 34,3% del ámbito del aprendizaje profundo hasta 2030².

El lenguaje de programación más popular actualmente es Python³⁴. Con este lenguaje de programación se ha creado un ecosistema de herramientas orientadas a la ciencia de datos que incluyen diversidad de bibliotecas que implemetan multitud de algoritmos de aprendizaje automático y profundo siguiendo los principios del software libre. La popularidad de Python ha convertido a este ecosistema en el preferido⁵ para el desarrollo de sistemas de aprendizaje automático. Estas tecnologías permiten usar algoritmos para estructurar datos, aprender de ellos y luego ser capaces de hacer una predicción o sugerencia en base a dichos datos.

1.1. Estado del arte

Una parte de los estudios que aplican el uso de relojes inteligentes para la detección de alteraciones de estados afectivos se aplica a la detección de estados de estrés [35]. En las siguientes secciones se describen cómo estas investigaciones abordan algunos de los aspectos fundamentales de estos procesos de detección.

1.1.1. Selección de dispositivos

Hoy en día existe un uso extendido de dispositivos electrónicos comerciales que sirven para monitorizar diferentes variables fisiológicas y que se sitúan para su funcionamiento en alguna parte del cuerpo. La gran accesibilidad a estos recursos permite estudiar la relación que existe entre estados emocionales y determinados valores de variables fisiológicas [20]. Dentro de este tipo de dispositivos podemos encontrar bandas, pulseras, relojes y anillos inteligentes. Aunque algunos de estos dispositivos disponen de una alta precisión y podrían equipararse a los dispositivos que están certificados para uso médico [39][46], no están incluidos dentro del grupo de dispositivos que se conocen como relojes inteligentes comerciales. También son utilizados en diferentes trabajos de detección de estrés dispositivos específicamente creados para el ámbito de la investigación [44][7][22], donde destacan

¹Mercado aprendizaje automático: <https://www.fortunebusinessinsights.com/machine-learning-market-102226>

²Mercado aprendizaje profundo: <https://www.grandviewresearch.com/industry-analysis/deep-learning-market>

³Índice TIOBE: <https://www.tiobe.com/tiobe-index/>

⁴<https://pypl.github.io/PYPL.html>

⁵<https://www.anaconda.com/state-of-data-science-2021>

los dispositivos creados por el fabricante Empatica⁶. Este tipo de dispositivo es capaz de captar una gran cantidad de variables fisiológicas y ofrecerlas para su estudio, pero son dispositivos de alto coste y no pueden clasificarse como relojes inteligentes comerciales. Para poder recoger el mayor número de variables fisiológicas posibles también son utilizadas estrategias de combinación de dispositivos [13]. Es posible utilizar exclusivamente los datos ofrecidos por relojes inteligentes comerciales, pero esto conlleva una reducción drástica de las variables de estudio disponibles [24] (frecuencia cardiaca y aceleración). La selección del dispositivo adecuado requiere un estudio de los modelos disponibles en el mercado. Es necesario evaluar diversas características, entre las que podemos destacar la disponibilidad de variables fisiológicas relacionadas con el estrés, la frecuencia de muestreo de dichas variables, si se puede acceder a los datos en tiempo real, si el usuario puede interactuar o recibir estímulos desde el propio reloj [45].

1.1.2. Señales

Las variables fisiológicas que están relacionadas con cambios en los estados afectivos son: la actividad electrodermal o conductancia eléctrica de la piel (EDA) [47] y la variabilidad del ritmo cardíaco (VRC) [31]. EDA puede reflejar procesos cognitivos y emocionales del sistema nervioso central, siendo aplicada a la detección de estados de ánimo y trastornos mentales [15][19]. La VRC corresponde al análisis de las variaciones de la frecuencia cardíaca (FC) o del intervalo de tiempo entre cada ciclo cardíaco consecutivo, es decir, el intervalo entre latidos (IBI). La medición de la VRC a corto plazo (aproximadamente 5 minutos) y a largo plazo (más de 24 horas) se ha demostrado eficaz para la medición de procesos reguladores o para predicción de patologías como el ataque al corazón, accidente cerebrovascular y otras causas de mortalidad. Diversos estudios relacionados con la psicología clínica y social realizan experimentos que miden la VRC en períodos de tiempo inferiores a 5 minutos[43]. Los relojes inteligentes incorporan sensores electrocardiográficos (ECG) y de fotopletismografía (PPG) que son capaces de medir la VRC y la FC. Estos sensores identifican el pico que la onda del pulso para determinar cuando un latido ha ocurrido. El procesamiento de los datos la señal VRC pueden realizarse mediante métodos en el dominio del tiempo o métodos en el dominio de la frecuencia [36][8][18].

1.1.3. Etiquetación de los datos

La recogida de datos para investigaciones en la detección de cambios en el estado afectivo (fundamentalmente en el estrés) mediante el uso de relojes inteligentes, se realiza mediante la exposición al usuario a diferentes situaciones donde se esperan respuestas afectivas también diferentes. Tras informar a los participantes del objetivo del experimento, se les expone a la realización de diferentes tipo de actividades como pueden ser acudir a clase, realizar una exposición o meditar. Este registro

⁶Empatica: <https://www.empatica.com/research/e4>

de actividades y momento en que se realizan también puede realizarse a posteriori si el experimento no dirige las actividades de los participantes, sino que monitoriza las actividades de la vida diaria. En todo caso, este registro temporal de las actividades conforman un primer nivel contextual para los datos recogidos. Para conocer el nivel de estrés de los participantes durante la realización de las actividades, en algunos de los estudios se realiza una valoración mediante formularios de evaluación de los que se extraerá una puntuación con la que se etiquetarán los valores de las series temporales recogidas. Para identificar el nivel de estrés de los participantes, se aplican cuestionarios de auto-evaluación como la escala de estrés percibido (PSS) [9] o el índice de carga mental (NASA-TLX) [23]. Estos cuestionarios se aplican para valorar el nivel basal previo al experimento y también el nivel de estrés alcanzado tras cada actividad. Algunos estudios combinan diferentes cuestionarios o adaptaciones de los mismos al contexto del experimento [6][7][22]. Debido a las capacidades de los relojes inteligentes, pueden incorporarse mecanismos simples de autoevaluación por parte del usuario a través de aplicaciones ejecutadas en el propio reloj [28] para indicar qué está sintiendo en un momento preciso. Toda esta información acerca de las emociones sentidas durante la realización de la actividad es utilizada para etiquetar los datos recogidos por los sensores de los relojes. La secuencia de datos recogida debe dividirse en pequeños fragmentos para poder aplicar algoritmos que analizan patrones en dichos fragmentos. Un aspecto que diversos autores destacan es la selección de un tamaño adecuado de segmento ya que éste influye en el rendimiento de los algoritmos aplicados [7][28][44].

1.1.4. Algoritmos de aprendizaje automático

Una vez que se dispone de un conjunto de datos proveniente del reloj inteligente, que posteriormente ha sido etiquetado, se selecciona un algoritmo que pueda realizar predicciones de niveles de estrés. Los trabajos de revisión muestran cómo en el ámbito de la detección de niveles de estrés mediante relojes inteligentes, el uso mayoritario de conjunto de datos etiquetados, conlleva el uso de algoritmos del ámbito del aprendizaje supervisado [26][5]. Los algoritmos de aprendizaje supervisado que mejor rendimiento muestran en diversos trabajos son las máquinas de vectores soporte (SVM), K vecinos más cercanos (KNN), regresión logística y bosques aleatorios (RF). A partir de estos algoritmos es posible combinarlos mediante técnicas de ensamblado con el objetivo de obtener mejores resultados[17].

1.2. Motivación

Las prestaciones e información que generan los relojes inteligentes actuales, junto con los algoritmos de inteligencia artificial, configuran una excelente plataforma para crear nuevas aplicaciones más allá del ámbito del seguimiento de la actividad física y de la salud. El presente proyecto está enmarcado dentro de Proyectos de Innovación Docente (PID) en el contexto del proyecto de investiga-

ción INT2AFF ENFOQUE DE DESARROLLO INTELIGENTE E INTRASUJETO PARA MEJORAR ACCIONES EN SISTEMAS ADAPTATIVOS EDUCATIVOS QUE CONSIDERAN EL ESTADO AFECTIVO (PGC2018-102279-B-I00). Este trabajo pretende estudiar las posibilidades de aplicación de estos recursos y técnicas a la detección de cambios en el estado afectivo de estudiantes durante la realización de la tarea Marco Lógico Colaborativo (MLC) [41]. MLC es un enfoque para mejorar el seguimiento de los estudiantes en escenarios de aprendizaje colaborativo estructurados en el que se pueden gestionar también de forma personalizada el estado afectivo de los estudiantes [40]. La implementación de una herramienta inteligente permite descargar al equipo docente de la gestión de la colaboración entre los estudiantes.

Los estudios sobre la aplicación de los relojes inteligentes a la detección de estados de estrés parten de la utilización de datos etiquetados. El proceso de etiquetación requiere la aplicación de cuestionarios de autoevaluación a lo largo del desarrollo de las actividades que realizan los usuarios. Esta recopilación de información requeriría establecer nuevas pautas y modificar el desarrollo habitual de las tareas de aprendizaje colaborativo según MLC. Existen un grupo de algoritmos dentro del ámbito del aprendizaje automático que no requieren una construcción previa de un conjunto de datos etiquetado. Estos algoritmos se engloban bajo la denominación de aprendizaje no supervisado y suelen estar orientados a realizar tareas de agrupamiento automático. La detección de alteraciones en el estado afectivo mediante este tipo de algoritmos permitiría una aplicación de los relojes inteligentes al proceso MLC sin la obligación de incorporar elementos que interrumpen la dinámica actualmente establecida.

Por otro lado, la utilización de algoritmos que no requieren de una elaboración previa de un conjunto de datos extraído de un reloj inteligente abre la posibilidad de aplicar en tiempo real los resultados que ofrezcan dichos algoritmos. Disponer de los datos de una forma inmediata, y realizar un procesado ágil, podría permitir dar información al propio usuario sobre el estado afectivo que está experimentando. Este mecanismo de retroalimentación puede convertirse en medio para la actuación sobre el usuario y una oportunidad también para la autoregulación del mismo.

Alcanzar una detección automatizada en tiempo real de alteraciones del estado afectivo facilitaría la incorporación de este tipo de tecnologías en el proceso MLC.

1.3. Objetivos

El objetivo principal es aplicar algoritmos de aprendizaje no supervisado a los datos extraídos de un reloj inteligente para la detección de cambios en el estado afectivo.

- Conocer los mecanismos de extracción de los datos generados por el reloj y automatizar su recogida.
- Estudiar diversos algoritmos de aprendizaje automático aplicables a los datos recogidos directamente del reloj sin ningún tipo de etiquetación o información previa.

- Desarrollar eficientemente una aplicación que permita evaluar el desempeño del modelo en tiempo real.

1.4. Fases del trabajo

En la primera fase del proyecto se realiza una evaluación de diferentes relojes inteligentes con el fin de delimitar las funcionalidades que proveen para adquirir un conjunto de datos mínimo. Este conjunto de datos debe maximizar el número de variables disponibles a través de sensores que puedan ayudar a la evaluación del estado afectivo, así como la granularidad temporal de los mismos.

Tras la selección de un modelo determinado se procede a la recogida de datos a través del reloj inteligente durante tres actividades diferentes. Para ello se desarrolla una aplicación específica que envía los datos que captan los sensores disponibles a un repositorio en la nube a través de una interfaz creada ad hoc. Dicha interfaz permite también la consulta de los datos para poder ser analizados desde entornos donde realizar un análisis de los mismos.

Debido a que los datos son series temporales, y no hay información a priori sobre los mismos, se aplican algoritmos de aprendizaje no supervisado de agrupamiento. El agrupamiento es la partición de n observaciones en k grupos, donde cada grupo se caracteriza por la homogeneidad (similitud de los elementos dentro de un grupo) y la separación (diferencia con los elementos de otros grupos). Se utilizan dos enfoques alternativos: algoritmos convencionales alimentados con series temporales modificadas y algoritmos que son capaces de procesar directamente las series temporales. La utilización de algoritmos convencionales conlleva la realización de un cambio de representación de los datos de las series temporales. Debe transformarse la secuencia de valores de la serie temporal a una representación diferente con la que puedan alimentarse los algoritmos. Usualmente se crean multitud de características sobre las que posteriormente se realiza una selección de las más relevantes para reducir de la dimensionalidad del conjunto de datos y que los algoritmos alcancen un mayor rendimiento[5]. Estos algoritmos convencionales están ampliamente documentados y permiten la utilización de las bibliotecas más utilizadas. Los algoritmos que procesan las series temporales sin que necesiten ser modificadas son algoritmos cuyas medidas de similitud evalúan la forma de las series temporales. Este comportamiento permite calcular adecuadamente la similitud entre series temporales que están afectadas por deformaciones como la traslación o el escalado. Son algoritmos específicos que no tienen aplicación fuera del ámbito de las series temporales, por lo que hay poca documentación y no pueden utilizarse las bibliotecas de aprendizaje automático que están ampliamente aceptadas por la comunidad.

Por último, se despliega el modelo creando una capa de visualización donde puede realizarse un seguimiento de la ejecución del algoritmo seleccionado en tiempo real, monitorizando el proceso completo desde el envío del dato desde la pulsera hasta la asignación de un grupo para la serie temporal por parte del algoritmo de agrupamiento.

Para la implementación de este proyecto se utilizará Python⁷, así como el ecosistema habitual de bibliotecas utilizadas para aprendizaje automático en el que destaca las bibliotecas Pandas, Numpy y Scikit-learn. El desarrollo de las distintas fases del análisis de los conjuntos de datos y rendimiento de los algoritmos se realizó en la plataforma *Google Colab*⁸. El despliegue de la interfaz de usuario con el algoritmo se realizará en la plataforma *Streamlit*⁹, la cual permite desarrollos en la nube orientados a implementaciones *low-code* en Python.

1.5. Estructura del documento

Tras la introducción del capítulo 1, en el capítulo 2 se inicia con el análisis de diversos relojes inteligentes con el fin de seleccionar aquel que permita extraer el conjunto de datos más adecuado. Posteriormente, se comparan diversos algoritmos de aprendizaje automático no supervisado y se seleccionan aquellos que han demostrado un mejor rendimiento. Tras el ajuste de algunos de sus hiperparámetros el capítulo se adentra en la descripción del despliegue del modelo para implementar su funcionamiento en tiempo real.

En el tercer capítulo se describen los resultados obtenidos para cada tipo de algoritmo. Finalmente, en el cuartocapítulo, se exponen las conclusiones del análisis realizado a los resultados y se sugieren líneas de trabajo futuro.

⁷Python: <https://www.python.org/>

⁸Google Colab: <https://colab.research.google.com/>

⁹<https://streamlit.io/>

Capítulo 2

Materiales y métodos

Este capítulo sigue en su desarrollo la estructura general de un proyecto de aprendizaje automático[21]. Una vez enmarcado el problema y analizado el estado del arte en el capítulo 1, el presente capítulo comenzará con la fase en la que se deben obtener los datos. En este proyecto la obtención se realizará a través del reloj inteligente durante la realización de una actividad. Una vez consolidado un conjunto de datos se procede a la exploración de los datos que los sensores del reloj han producido, y que conformarán las series temporales de partida. Posteriormente se realiza la preparación de los datos para poder alimentar los modelos que van a ser probados. Tras las pruebas iniciales se seleccionan los modelos candidatos con mejor desempeño y se procede a un ajuste de los hiperparámetros. Se selecciona el modelo más idóneo tras la fase de ajuste. Por último, se prepara una solución que será desplegada. Todas estas etapas y los elementos tecnológicos más reseñables aparecen en la figura 2.1.

2.1. Obtención de datos

Los modelos más avanzados de relojes inteligentes incorporan una multitud de sensores, algunos de los cuales miden variables fisiológicas que están relacionadas con los cambios afectivos. Este tipo de dispositivos contienen sensores de interés para el objetivo de este trabajo:

- Sensor de actividad electrodermal: es capaz de medir con precisión cambios en las propiedades eléctricas de la piel. Estos cambios son causados por alteraciones en la secreción de sudor y la actividad de las glándulas sudoríparas como resultado de cambios en la actividad del sistema nervioso simpático. La localización adecuada para la medición de esta variable es la palma de la mano. Este sensor también registra las señales eléctricas que emite el latido cardíaco.
- Oxímetro: es un sensor que contiene unos LEDs que emiten una luz roja con una longitud de onda de 660 nanómetros y una luz infrarroja con una longitud de onda de 940 nanómetros. El motivo es que la sangre oxigenada absorbe la radiación infrarroja y la sangre desoxigenada

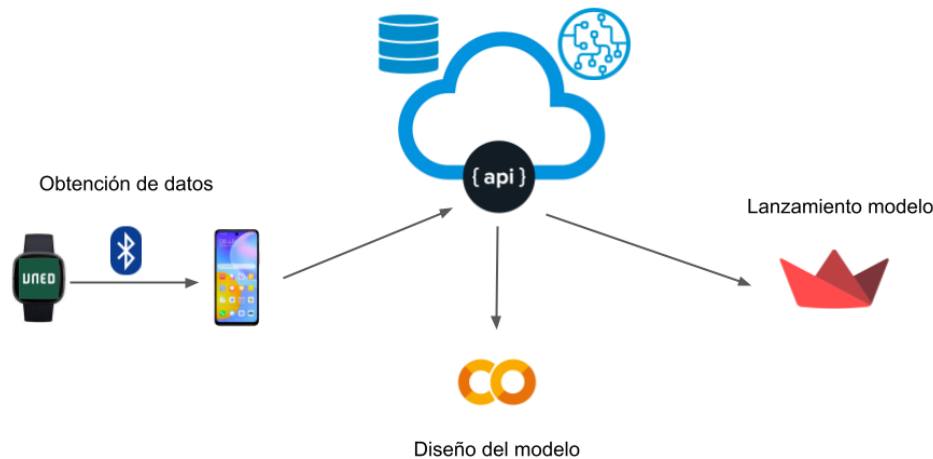


Figura 2.1: Se inicia el proyecto con el desarrollo de una app que se ejecuta en el reloj y teléfonos inteligentes para extraer los datos. A través de una API los datos son almacenados y transformados al formato adecuado para su posterior procesamiento. La fase de diseño comprende la prueba de diversos modelos en la plataforma *Google Colab*. Por último, a través de la se provee de datos para alimentar a la aplicación web desplegada en la plataforma *Streamlit*.

absorbe la luz roja. En la parte inferior del oxímetro hay un fotodiodo que recibe la luz no absorbida, y así se puede estimar de forma indirecta el nivel de oxígeno en sangre.

- Sensor de temperatura cutánea: a través de la temperatura de la piel de la muñeca que está en contacto con el reloj se obtiene esta variable fisiológica.
- Sensor del ritmo cardíaco: para determinar el ritmo cardíaco el sensor parpadea con sus luces LED verdes muchas veces por segundo y utiliza fotodiodos sensibles a la luz para detectar estos cambios de volumen en los capilares por encima de la muñeca. A continuación, el dispositivo calcula cuántas veces late el corazón por minuto. Se utilizan luces LED verdes porque maximizan la señal detectada en los capilares cerca de la superficie de la piel.

Los fabricantes de estos dispositivos pueden proveer mecanismos para que los usuarios se descarguen los datos registrados mediante el uso de sus relojes, para darles una utilidad específica:

- Acceso a la web del usuario en la que puede realizarse la descarga manual de los datos
- Acceso a una API para poder implementar una descarga automatizada de datos consolidados

Tanto el número de variables, como la granularidad temporal de las mediciones de los datos ofrecidos por estas interfaces son elementos fundamentales para valorar su idoneidad y disponibilidad de los datos.

El primer modelo que se evalúa es el reloj inteligente Galaxy Watch 4 del fabricante Samsung. Es un reloj orientado al seguimiento de la salud de los más avanzados del mercado y contiene un grupo de sensores que es capaz de medir (entre otras variables) FC, VRC, el nivel de oxígeno en sangre y la presión arterial. Tras evaluar los requerimientos del dispositivo se evidencian algunos elementos limitantes:

- Como la mayor parte de los relojes inteligentes, están diseñados como un periférico del teléfono inteligente, por ello es necesario instalar una aplicación en el teléfono inteligente con la que se realiza la sincronización y actuación sobre el reloj. En el caso del reloj de Samsung es necesario realizar una instalación en un teléfono inteligente que tenga los servicios de Google (GMS) y también la certificación de Google. Por lo tanto, en dispositivos de proveedores como Huawei, Apple, Xiaomi, Vivo, etc no es posible utilizar este reloj.
- Si se opta por configurar el reloj con un teléfono inteligente que sí cumpla los requisitos del punto anterior pero no sea del fabricante Samsung, no se puede acceder a las variables del intervalo entre latidos y la tensión arterial.
- El fabricante Samsung no dispone de una API para la descarga de los datos offline de una forma automatizada.
- La creación de una aplicación específica para el reloj que acceda al resto de variables requiere participar en un programa como partner para acceder a Samsung Privileged Health SDK.

Samsung como fabricante implementa un enfoque muy orientado hacia la utilización de su ecosistema de dispositivos y aplicaciones, sin las cuales es inviable la utilización del reloj Galaxy Watch 4 para los fines que este proyecto persigue.

Seguidamente se evalúa el reloj inteligente Sense del fabricante Fitbit. Este reloj también es un modelo orientado al cuidado de la salud que dispone entre otras, de las variables FC, VRC, nivel de oxígeno en sangre, actividad electrodermal y temperatura. Este reloj puede ser asociado tanto a dispositivos Android como iOS. Fitbit provee de una API para la descarga automatizada de los datos consolidados. Tras implementar la descarga mediante la API se observa que los datos disponibles tienen una resolución temporal máxima de un minuto. Disponer de un valor por minuto cuando las actividades a monitorizar tienen una duración aproximada de una hora, genera series temporales con muy pocos elementos. A esta baja granularidad de los datos se une que la API únicamente ofrece un subconjunto de las variables que el dispositivo ofrece en su interfaz para aplicaciones. Por estos motivos se descarta la utilización de los mecanismos de descarga de datos mediante la API para el desarrollo del presente trabajo.

Un enfoque alternativo a la utilización de las interfaces provistas por el fabricante es la creación de una app que extraiga los datos desde el propio dispositivo. Esta orientación posibilita una mayor granularidad temporal de los datos y el procesamiento de los datos en tiempo real mientras se monitoriza la actividad a realizar. Fitbit dispone, para el desarrollo de aplicaciones, tanto de un entorno

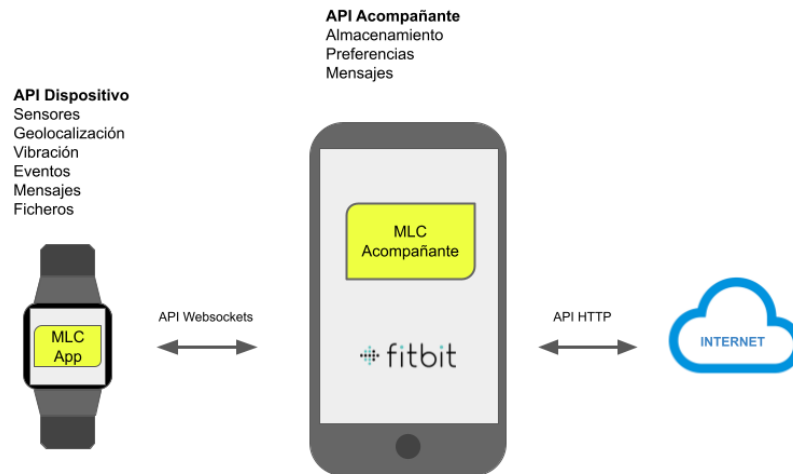


Figura 2.2: Arquitectura de la aplicación Fitbit

web (Fitbit Studio¹), como de una herramienta de línea de comandos integrable con editores de código y sistemas de control de versión. Estas herramientas están disponibles simplemente mediante el registro de usuario en su plataforma.

La estructura de una aplicación está compuesta por dos elementos bien diferenciados:

- La lógica de la aplicación que se ejecuta en el propio reloj inteligente y que permite la lectura de los sensores, la interacción con la capa de presentación y la comunicación con el teléfono inteligente.
- La lógica de la aplicación que se ejecuta en el teléfono inteligente en el contexto de la propia aplicación de Fitbit. Tiene la capacidad de realizar peticiones a Internet y comunicarse con la aplicación del reloj inteligente.

En la figura 2.2 pueden observarse los dos componentes de la app MLC desarrollada junto con los principales componentes que contienen las API disponibles en el ámbito del reloj y teléfonos inteligentes.

Todas estas características hacen del modelo Sense de Fitbit un dispositivo más adecuado, por lo que será seleccionado para la implementación del resto etapas del proyecto.

Los módulos de ambas lógicas de aplicación se implementan en JavaScript tal como define la guía de desarrollo del fabricante². La aplicación resultante ha sido registrada en la galería de aplicaciones de Fitbit para poder ser instalada en cualquier otro dispositivo Sense mediante el envío de un enlace³

¹Fitbit Studio: <https://studio.fitbit.com/>

²SDK: <https://dev.fitbit.com/build/guides/user-interface/javascript/>

³<https://gallery.fitbit.com/details/13bce84b-8296-4a7c-b990-79cb847d3df6>

de instalación. La aplicación reside en la plataforma de Fitbit por lo que es necesario cumplir con sus requerimientos en relación al uso y licencia de la aplicación⁴.

La aplicación desarrollada lee los sensores disponibles a través de la API del dispositivo cada segundo. El API del dispositivo sólo publica la frecuencia cardíaca como variable de interés fisiológico. La frecuencia cardíaca viene expresada en latidos por minuto. El resto de variables sólo están disponible mediante la visualización en la aplicación del teléfono inteligente. No existe ninguna variable en el dispositivo que lo identifique unívocamente. Una vez leídos los sensores se empaqueta la información en formato JSON y se realiza una llamada desde la lógica de la aplicación que reside en el teléfono inteligente a un método de una API creada para su almacenamiento en formato CSV. Esta API está alojada en un servicio en la nube específico para alojar módulos de Python. De esta forma se consigue construir un dataset con las variables de frecuencia cardíaca y aceleración 3D (medida en m/s^2) con una resolución temporal de un segundo.

La interfaz de la aplicación del reloj guía al usuario por dos fases diferenciadas: fase de calibrado y fase de monitorización. Durante la fase de calibrado se solicita al usuario que esté sentado, tranquilo y respire calmadamente durante 5 minutos. Una vez finalizado este tiempo se requiere al usuario que inicie la fase de monitorización donde abordará la realización de una actividad. Podría realizarse cualquier actividad que se desee, pero en este trabajo vamos a realizar tareas donde el individuo esté sentado y afronte una tarea de resolución de problemas fundamentalmente. El usuario puede detener la monitorización de la actividad en cualquier momento. La secuencia de pantallas que conforman la interfaz gráfica de la aplicación del reloj inteligente puede verse en la figura 2.3.

El flujo continuo de datos desde el reloj al servidor de almacenamiento conlleva un uso intensivo de la batería del dispositivo y que la sincronización con el teléfono inteligente presente desconexiones durante la duración de la actividad.

2.2. Exploración

Los datos obtenidos mediante el proceso anteriormente descrito conforman una serie temporal. Una secuencia compuesta de una serie de símbolos pertenecientes a un alfabeto particular es usualmente denominado secuencia temporal, y una secuencia de elementos que poseen un valor, se denomina serie temporal. Una serie temporal es un dato dinámico ya que los valores de sus propiedades cambian en función del tiempo, lo cual conlleva que los valores de cada elemento de una serie temporal está compuesto por una o más observaciones realizadas cronológicamente. Por lo tanto, los datos de series temporales también son denominados datos dinámicos.

Se han creado para este proyecto tres series temporales diferentes para evaluar los modelos de aprendizaje automático. Para ello se ha realizado (tras la fase de calibrado) una tarea durante 60 minutos. Las tres series temporales han sido registradas en tres situaciones diferentes:

⁴<https://dev.fitbit.com/legal/platform-terms-of-service/>

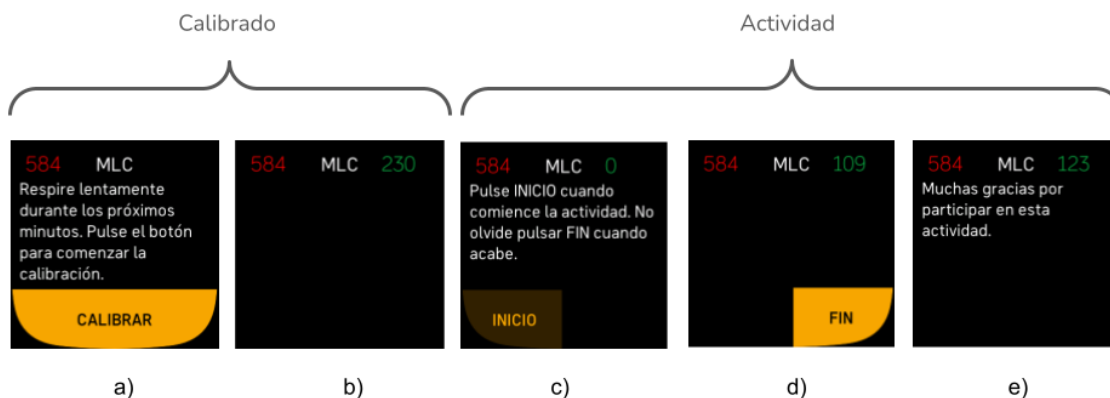


Figura 2.3: La secuencia comienza con el inicio del calibrado para establecer un nivel basal del usuario (a). El calibrado se desarrolla durante una cuenta atrás de 300 segundos visibles en el contador verde de la esquina superior derecha (b). La pantalla de inicio de la actividad (c) dará paso al envío de mediciones cada segundo hasta que se pulse el botón (d), tras lo cual aparecerá un agradecimiento (e).

- Resolución de un cubo Rubik mientras suena música estresante.
- Resolución del mayor número de problemas posibles del desafío anual de programación *Advent of Code*⁵ mientras suena música estresante.
- Visualización de episodio de serie de televisión dramática.

Las tres actividades se realizaron por la misma persona durante en un mismo día. Las dos primeras tareas se diseñaron con el fin de crear una actividad que pudiera generar una situación de estrés ante la resolución de un problema. El participante no tenía información previa de cómo resolver ambos problemas. Para incrementar la intensidad de la dificultad se introdujo un música de carácter estresante a alto volumen durante la realización de toda la actividad.

La realización de la tarea de visualización de un episodio dramático tiene como finalidad incorporar el registro de una actividad donde no se esperan situaciones de estrés sino un rango de emociones diferentes como pueden ser la tristeza o sorpresa.

Las tres actividades se realizaron con el usuario sentado en reposo para evitar que el movimiento físico afectara el registro de la FC.

Con estas tres tareas se pretende construir un conjunto de datos donde se viven por el usuario situaciones de estrés, así como otro tipo de emociones que también pueden generar cambios en el

⁵*Advent of Code*: <https://adventofcode.com/>

índice	tiempo	id	etapa	fc	accx	accy	accz
0	10/05/2022 12:53:53.132862	45	1	71	0.890670	0.387872	10.249889
1	10/05/2022 12:53:54.462653	45	1	72	0.482446	-3.160442	9.170072
2	10/05/2022 12:53:55.079409	45	1	74	0.754196	-3.275368	9.065921
3	10/05/2022 12:53:56.219785	45	1	74	0.682368	-3.342407	9.172466
4	10/05/2022 12:53:57.136063	45	1	73	0.699128	-3.505218	9.112609
...

Tabla 2.1: Fragmento de serie temporal

estado afectivo.

El conjunto de datos se almacena en formato CSV conforme se reciben las medidas que envía el dispositivo. En el cuadro 2.1 puede observarse un fragmento extraído del conjunto de datos de la serie temporal Rubik. Puede observarse en la variable *tiempo* la cadencia de un segundo entre mediciones. La columna *id* indica el número de serie temporal al que pertenecen todos los registros. Mediante el campo *etapa* se indica si la medida pertenece a la fase de calibrado o de monitorización. La variable *fc* indica la frecuencia cardíaca en ese instante. Las variables *accx*, *accy* y *accz* son las tres componentes de la aceleración.

Se realiza la exploración de datos en el entorno *Google Colab* partir de la descarga directa de los datos a través de uno de los métodos de la API desarrollada. Las series temporales almacenadas contienen cada una más de 3500 mediciones entre la fase de calibrado y la monitorización. Las medidas están registradas en intervalos constantes de aproximadamente un segundo, por lo que no es necesario realizar ningún proceso de muestreo o regularización sobre los datos. En la figura 2.4 se representan las tres series temporales que van a utilizarse para las siguientes etapas del trabajo y que han sido extraídas de los valores que proporciona el sensor de frecuencia cardíaca del reloj inteligente.

La exploración realizada sobre las series permite determinar:

- No existen valores nulos en ninguna de las variables medidas.
- Todas las medidas de la frecuencia cardíaca oscilan en el rango de 54 y 92 pulsaciones por minuto.
- No existe correlación entre ninguna de las componentes de la aceleración (ni su módulo) con la variable frecuencia cardíaca. Todas las actividades se realizaron con el usuario del reloj sentado en una silla.

Para cada actividad se obtiene una única serie temporal que contiene todas las mediciones que se han realizado en un período de tiempo. Como el objetivo del proyecto es la detección de alteraciones del estado afectivo durante algún momento de la realización de la actividad, será necesario identificar si un fragmento de tiempo de la actividad tiene algún patrón que indica alteración. Para las series

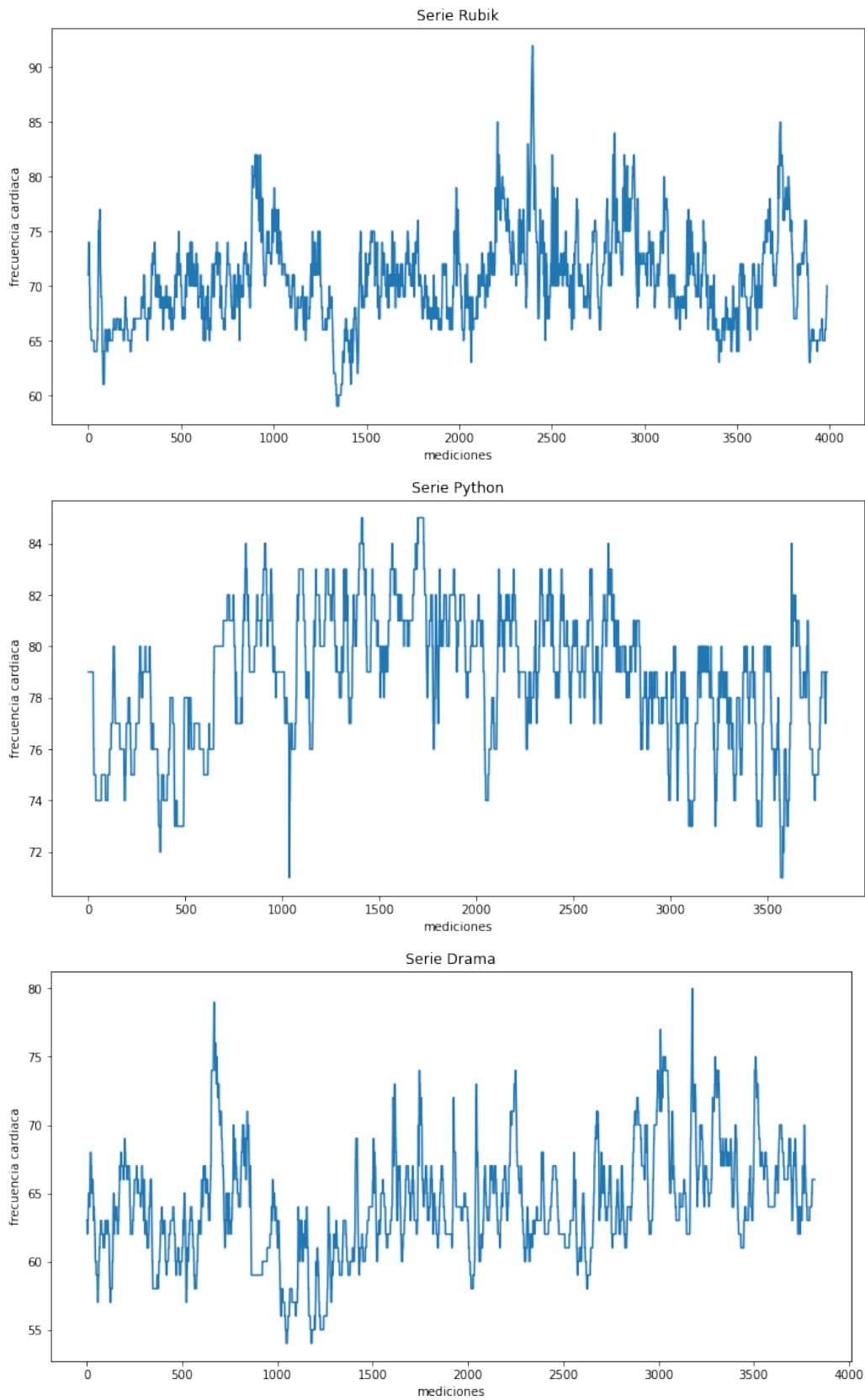


Figura 2.4: Series temporales de la variable frecuencia cardíaca registradas durante diferentes actividades

temporales generadas no existe información a priori, ni ningún tipo de etiquetación sobre las mediciones, por lo que el abordaje elegido es la utilización de algoritmos de agrupamiento [49], enmarcados dentro del área de aprendizaje no supervisado. Una taxonomía [1] de los trabajos publicados de agrupamiento de series temporales indica que hay tres categorías:

- Agrupamiento de serie temporal completa: aplicación de algoritmos convencionales a objetos que son series temporales.
- Agrupamiento de subsecuencias: agrupar segmentos de series temporales que son generados mediante una ventana temporal que se desliza. Por lo tanto, en este caso se agrupan fragmentos que provienen de una serie temporal larga.
- Agrupamiento de puntos de tiempo: es una aproximación similar al agrupamiento de subsecuencias, sin embargo, no todos los puntos de un segmento son asignados a un grupo si por ejemplo, algunos de ellos son considerados como ruido.

Por la naturaleza del problema ya descrito de la detección de un patrón en un fragmento de la serie temporal completa, el proyecto se engloba en la categoría de agrupamiento de subsecuencias, donde se extraen un conjunto de subsecuencias o segmentos mediante una ventana temporal de un tamaño definido. De esta forma, se realizará el agrupamiento de los segmentos extraídos de la serie total.

La naturaleza de las variables sujetas a estudio y la fragmentación de las series temporales en segmentos de reducido tamaño conlleva que no se realice el estudio de las componentes de las series temporales originales. Las componentes de la serie temporal denominadas tendencia, cíclica y estacional pretenden detectar comportamientos a lo largo de la serie temporal completa, mientras en este trabajo se enfoca en encontrar patrones en los segmentos de las series temporales que permitan agruparlos por similitud[7][28][44].

2.3. Preparación

A la hora de seleccionar los algoritmos de agrupamiento podemos destacar tres aproximaciones diferentes [49]: basados en series temporales, basados en características y basados en modelos. Los algoritmos basados en series temporales son algoritmos clásicos de agrupamiento a los que se les ha modificado la medida de similitud para que pueda ser aplicada a datos dinámicos directamente. La aproximación de los algoritmos basados en características y basados en modelos procesan la serie temporal para generar una nueva representación, que generalmente es un vector de características de menor dimensión. Posteriormente se aplican algoritmos convencionales para datos estáticos a la nueva representación. Existen multitud de métodos de representación de series temporales [11] que permiten una reducción de la dimensionalidad. Debido al reducido tamaño de los segmentos que se van a agrupar, se ha utilizado una representación que aplica indicadores estadísticos a cada segmento. Para generar estos indicadores de cada segmento se ha utilizado una biblioteca específica para ello

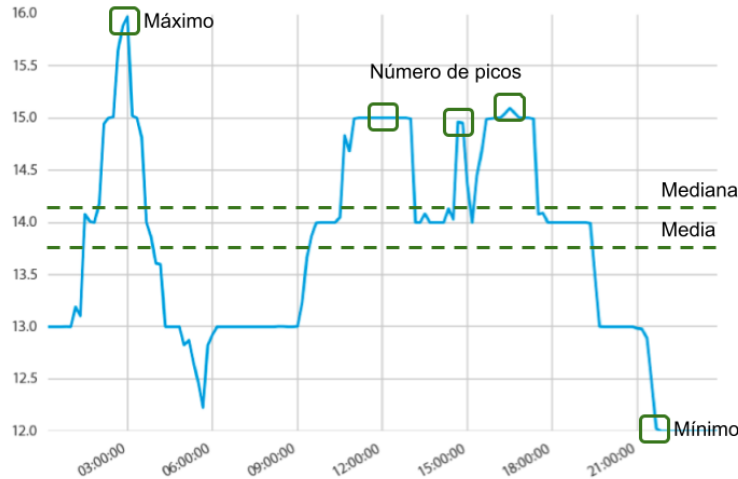


Figura 2.5: Generación de nueva representación para una serie temporal a partir de la extracción de características de la serie mediante indicadores. Tsfresh permite extraer más de cien características diferentes de una serie temporal.

denominada tsfresh⁶. Este herramienta extrae multitud de características y también permite filtrar automáticamente aquellas que realmente son relevantes. La figura 2.5 muestra un ejemplo de cómo tsfresh realiza la representación de una serie temporal mediante el cálculo de nuevas características.

Para que los algoritmos que van a probarse muestren el mejor rendimiento, es importante que se aplique a los datos el escalado como transformación. Estos algoritmos ven mermados sus resultados cuando los datos de las variables de entrada están en muy diferentes escalas. Hay dos formas habituales de conseguir que todos los atributos tenga la misma escala: el escalado min/max y la normalización. En el escalado min/max a los valores de una variable se le sustrae el valor mínimo y se les divide por la diferencia entre el valor máximo y el valor mínimo, lo que da un resultado en un rango entre 0 y 1. La normalización procesa los datos de una variable sustrayendo el valor medio y dividiéndolo por la desviación estándar. La normalización no da como resultado valores en el intervalo entre 0 y 1. Se elige la normalización como método de escalado para este trabajo ya que es un método más robusto ante valores atípicos y no se requieren valores entre 0 y 1 por parte de los algoritmos que van a probarse.

Es importante señalar que para los algoritmos que no requieren cambio de representación, el escalado debe realizarse de tal forma que cada segmento es escalado individualmente. Este proceso permite que el rango de valores de un segmento no sea relevante, sino que la comparación entre segmentos pueda realizarse atendiendo únicamente a la forma de la secuencia de valores.

El proceso de segmentación de la serie temporal original, el cambio de representación y el escalado

⁶tsfresh: <https://tsfresh.readthedocs.io/>

han sido implementados mediante un objeto *pipeline* de la biblioteca scikit-learn⁷ con el fin de estructurar y reutilizar el código utilizado para la preparación y generación de características.

2.4. Prueba de modelos

Para la realización de las pruebas se seleccionan modelos que directamente utilizan los datos dinámicos, y modelos que requieren un cambio de representación previo para poder ser utilizados. Los modelos que requieren cambio de representación son los que están disponibles en la biblioteca scikit-learn para realizar tareas de agrupamiento. Estos algoritmos pueden ser clasificados por taxonomía[50] como:

- Algoritmos de partición: K-Medias, propagación por afinidad y agrupamiento espectral.
- Algoritmos jerárquicos: BIRCH, agrupamiento por aglomeración y Mean-Shift.
- Algoritmos de densidad: DBSCAN

Como algoritmos que procesan directamente datos dinámicos se han seleccionado K-medias con DTW y K-Shape. Todos estos modelos mencionados serán descritos brevemente junto con los resultados de las pruebas realizadas. Las pruebas se realizaron con segmentos de 10 segundos de duración. Esta longitud de segmento es la que mejor puntuación mostró en una exploración inicial del rendimiento del algoritmo K-medias con ventanas de tiempo entre 10 y 120 segundos, en incrementos de 10 segundos. En la sección de ajuste, se evaluarán diferentes tamaños de segmento para todos los algoritmos que superen la fase de pruebas. En el caso de que se pueda indicar como hiperparámetro al modelo, se determinaron 3 grupos como resultado esperado.

2.4.1. K-Medias

El algoritmo K-Medias[27] es un algoritmo simple capaz de agrupar conjuntos de datos con rapidez y eficacia, a menudo en pocas iteraciones. K-Medias parte de la idea de considerar el centro de un conjunto de datos como el centro del grupo (centroides). Es necesario indicar al algoritmo el número de grupos que deseamos obtener. Se comienza colocando los centroides de manera aleatoria (puede elegirse un número de instancias al azar igual al número de grupos a alcanzar y determinar que son los centroides iniciales). A continuación, se etiquetan las instancias en función del centroide más cercano y se vuelven a calcular los centroides. Este proceso se repite hasta que los centroides dejen de cambiar de posición. Está garantizado que el algoritmo convergerá en un número finito de pasos, aunque puede que no converja en la solución adecuada, sino que converja en un óptimo local. La complejidad computacional del algoritmo es por lo general lineal respecto al número de instancias,

⁷scikit-learn: <https://scikit-learn.org/stable/>

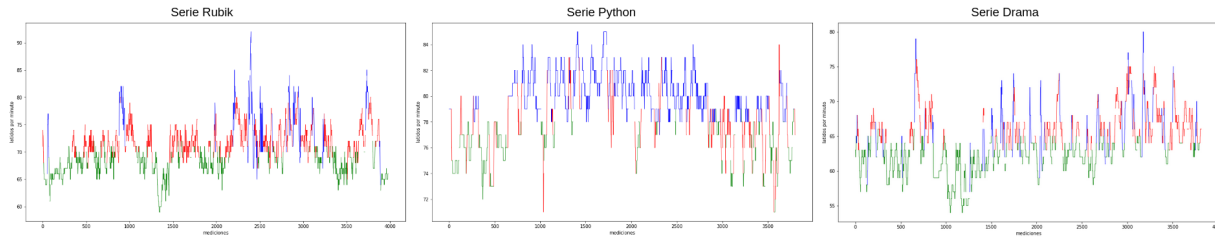


Figura 2.6: Resultado de agrupar en 3 grupos los segmentos de 10 segundos de duración de la FC. Se observa cómo los grupos se distribuyen por zonas horizontalmente.

el número de grupos y el número de dimensiones. Esto sólo es cierto cuando los datos tienen una estructura de agrupamiento, si no fuera así, tendería a una complejidad exponencial.

El algoritmo K-Medias resuelve un problema de optimización, siendo la función a optimizar o minimizar la suma de las distancias cuadráticas de cada objeto al centroide de su grupo. Los elementos del conjunto de datos se representan con vectores reales de d dimensiones (x_1, x_2, \dots, x_n) y el algoritmo construye k grupos donde se minimiza la suma de distancias de los objetos, dentro de cada grupo $S = \{S_1, S_2, \dots, S_k\}$, a su centroide. El problema se puede formular de la siguiente forma:

$$\min_S E(\mu_i) = \min_S \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

donde S es el conjunto de datos cuyos elementos son los objetos x_j representados por vectores, donde cada uno de sus elementos representa una característica o atributo. Se obtendrán k grupos o clusters con su correspondiente centroide μ_i .

En la tabla 2.6 puede observarse el resultado del agrupamiento de los segmentos de las tres series temporales.

En los resultados del algoritmo K-Medias se aprecia una definición de grupos que se visualizan a modo de bandas horizontales de similar altura. Este algoritmo utiliza como medida de similitud la distancia euclídea, por lo que es esperable un resultado como éste cuando el conjunto de datos no contiene zonas de muy diferente densidad, es decir, los datos no tienen forma de grupos sino que los datos están representados de una forma continua a lo largo del rango de valores.

2.4.2. Propagación por afinidad

El algoritmo de propagación por afinidad[25] está basado en grafos, por lo que no procesa distancias entre elementos. Para ello utiliza una matriz de similaridad, de tal forma que los puntos que están cerca entre sí tienen una alta similaridad y los más alejados se parecen poco. El algoritmo trabaja con dos matrices distintas:

- Matriz de responsabilidad: es la encargada de definir cómo de responsable es cada elemento,

por lo tanto, define la importancia que tiene.

- Matriz de disponibilidad: determina la cantidad de elementos vecinos que tiene un determinado elemento del conjunto de datos.

A partir de estas matrices se seleccionan los elementos que serán el centro de los grupos. Los elementos con mayor responsabilidad y disponibilidad serán los elegidos.

Dado un conjunto de datos con vectores reales de d dimensiones (x_1, x_2, \dots, x_n) s es una matriz $N \times N$ tal que $s(i, j)$ representa la similitud entre d_i y d_j . La similitud entre dos puntos es:

$$s(i, j) = -\|x_i - x_j\|^2$$

La matriz de responsabilidad R y la matriz de disponibilidad A se inicializan a cero. El algoritmo actualiza las matrices siguiendo un proceso iterativo:

- Matriz de responsabilidad

$$r(i, j) \leftarrow s(i, j) - \max_{j' \neq j} \{a(i, j') + (s(i, j'))\}$$

- Matriz de disponibilidad

$$a(i, j) \leftarrow \min \left(0, r(j, j) + \sum_i \max(0, r(i', j)) \right), i \neq j$$

$$a(j, j) \leftarrow \sum_{i' \neq j} \max(0, r(i', k))$$

En las pruebas de este algoritmo no se obtiene convergencia en alguna de las series o bien no se obtienen el número adecuado de grupos ya que no se definen a priori mediante un hiperparámetro. Por ello este algoritmo queda descartado para las siguientes fases del trabajo.

2.4.3. Agrupamiento espectral

El agrupamiento espectral[2] utiliza el espectro o valores propios de la matriz de similitud de los datos para realizar la reducción de dimensionalidad antes de agruparlos en menos dimensiones. La matriz de similitud se proporciona como entrada y consiste en una evaluación cuantitativa de la similitud relativa de cada par de elementos del conjunto de datos. Dado un conjunto de elementos, la matriz de similitud puede definirse como una matriz simétrica A , donde $A_{ij} \geq 0$ representa una medida de la similitud entre elementos del conjunto de datos con índices i y j . El enfoque general para el agrupamiento espectral es utilizar un método de agrupación conocido en los vectores propios relevantes de una matriz laplaciana de A . Los autovectores que son relevantes son los que corresponden a los autovalores más pequeños del Laplaciano, salvo aquellos que tengan valor 0.

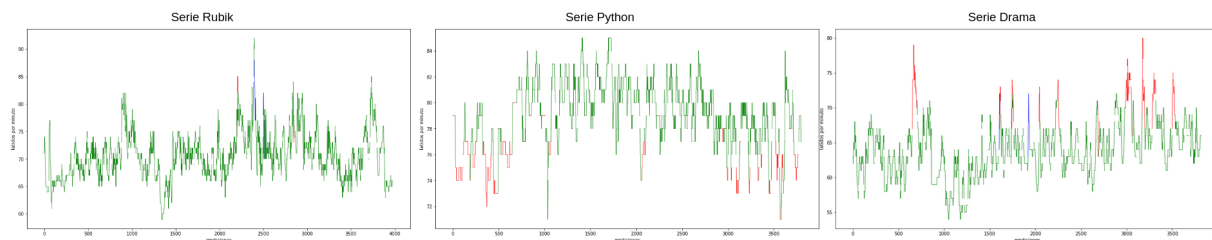


Figura 2.7: En las tres gráficas puede verse que hay un grupo mayoritario en componentes y dos grupos de tamaño muy reducido.

El algoritmo básico sigue los siguientes pasos:

1. Calcular el Laplaciano L .
2. Calcular el primer k de autovectores (los autovectores correspondientes a los k valores propios más pequeños de L).
3. Considerar la matriz formada por los primeros k vectores propios.
4. Agrupar los elementos en función de estas características.

Puede verse en la figura el resultado de las pruebas con el modelo de agrupamiento espectral.

El algoritmo agrupa la mayor parte de los segmentos en un grupo, que coincide con la franja donde hay más densidad de elementos. En los otros dos grupos quedan aquellos segmentos que forman picos o valles alejados de esa franja del grupo mayoritario.

2.4.4. BIRCH

El algoritmo BIRCH[51] puede agrupar grandes conjuntos de datos generando primero un resumen pequeño y compacto que contiene la mayor cantidad de información. Este resumen de los datos originales es el que realmente se agrupa. BIRCH es un método basado en agrupamiento jerárquico y por lo tanto, sólo necesita procesar una única vez el conjunto de datos. Las grandes regiones de datos son resumidas en regiones más pequeñas y densas llamadas entradas de características de agrupamiento. Una característica de agrupamiento se define como una tripleta (N, LS, SS) , donde N es el número de elementos de datos del grupo, LS es la suma lineal de los elementos y SS es la suma al cuadrado de los elementos del grupo. Asimismo el algoritmo utiliza una estructura de árbol para crear los grupos. El árbol está balanceado en altura almacenando las características de agrupamiento con la información necesaria para realizar un posterior agrupamiento jerárquico sin necesidad de manejar el conjunto de datos total. La figura 2.8 muestra cómo resuelve la tarea de agrupamiento el algoritmo BIRCH en las tres series temporales, determinando grupos que se distribuyen según el rango de valores de la variable de estudio. Estos resultados son similares a K-Medias, ya que utiliza para medir la distancia entre grupos la distancia euclídea entre centroides.

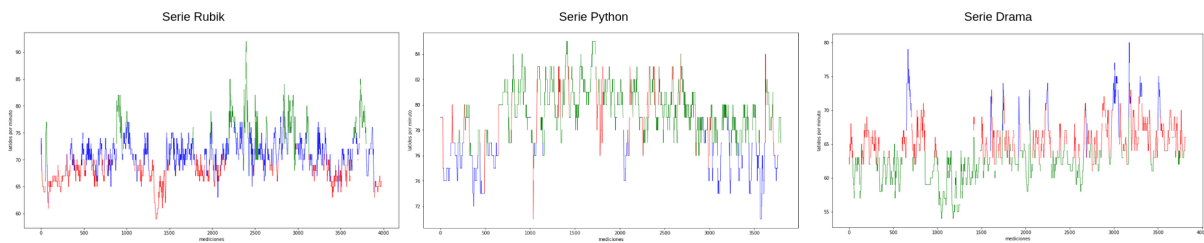


Figura 2.8: Prueba del modelo de agrupamiento BIRCH.

2.4.5. Agrupamiento por aglomeración

En el modelo de agrupamiento por aglomeración[32] o enfoque ascendente cada elemento del conjunto de datos es tratado como un único grupo y luego se fusionan sucesivamente pares de grupos hasta que todos los grupos se hayan fusionado en uno solo. El algoritmo básico se puede describir con los siguientes pasos:

1. Calcular la matriz de proximidad
2. Determinar que cada elemento del conjunto de datos sea un grupo
3. Repetir fusión de los grupos más cercanos y actualización de matriz de proximidad hasta que sólo quede un único grupo.

Para encontrar qué grupos son los más cercanos es necesario definir una medida de distancia entre los grupos. Pueden utilizarse cualquiera de las siguientes medidas:

- Conexión completa: la distancia se mide entre los dos puntos más lejanos de cada grupo.
- Conexión simple: la distancia es la distancia mínima entre dos grupos.
- Distancia entre medias: se calcula como la distancia entre las medias de cada uno de los dos grupos.
- La distancia promedio entre pares: el promedio entre todas las distancias que se pueden obtener entre todos los pares de elementos de ambos grupos.

En las pruebas realizadas se han utilizado los criterios de fusión descritos anteriormente, obteniendo resultados muy diversos. Los métodos de conexión completa, simple y entre medias determinan un gran grupo donde se concentran la mayor parte de los segmentos y agrupan en los otros dos grupos los segmentos que forman los picos de mayor magnitud, es decir, los elementos más alejados del grupo general. En la figura 2.9 se muestran los resultados obtenidos mediante el criterio de fusión de distancia promedio entre pares o también denominado enlace Ward [48]. Con este tipo de enlace el algoritmo agrupa con el patrón de bandas horizontales distribuyendo uniformemente el rango de valores entre los tres grupos. Este comportamiento es similar a K-Medias ya que de igual forma pretende minimizar la varianza de los grupos y utiliza para ello la medición de la distancia euclídea.

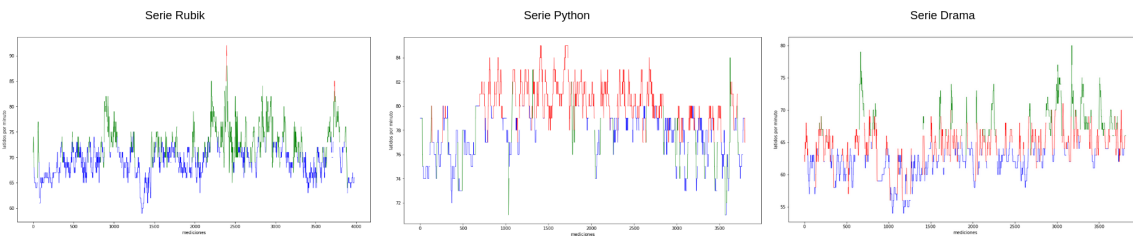


Figura 2.9: Prueba del modelo de agrupamiento aglomerativo.

2.4.6. Mean-Shift

Mean-Shift [12] es un algoritmo que utiliza ventanas deslizantes para encontrar áreas de elementos con alta densidad. Es también un modelo basado en centroides, por lo que se localizan los puntos centrales de cada grupo mediante el cálculo del elemento central de la ventana deslizante. Las ventanas deslizantes son filtradas posteriormente para desechar duplicados cercanos. El algoritmo puede explicarse brevemente en los siguientes pasos suponiendo un conjunto de datos que forma un espacio bidimensional:

1. Comienza el algoritmo con una ventana circular deslizante centrada en un elemento seleccionado aleatoriamente y con un radio r . Mean-Shift realiza un desplazamiento iterativo de esa ventana a una región de mayor densidad en cada paso hasta converger.
2. En cada iteración el centro de la ventana se desplaza a regiones de mayor densidad, recalculando dicho centro como el valor medio de los elementos que engloba la ventana.
3. La ventana se desplaza hasta que no hay dirección en la que un desplazamiento pueda acomodar más puntos dentro del núcleo.
4. Los pasos 1 a 3 se realizan con un gran número de ventanas hasta que todos los elementos del conjunto de datos se encuentran dentro de alguna ventana. Cuando se solapan varias ventanas, se conserva la ventana que contiene mayor cantidad de elementos. Finalmente, los elementos quedan agrupados según la ventana en la que residen.

En este algoritmo no es posible definir el número de grupos mediante un hiperparámetro. La imposibilidad de definir un número de grupos unido a que en los conjuntos de datos no existen zonas claramente diferenciadas en densidad conlleva que se obtengan agrupaciones donde o bien se agrupan casi todos los segmentos en un grupo, o bien se genera un excesivo número de grupos. Existe un hiperparámetro que afecta al tamaño de las ventanas deslizantes denominado *ancho de banda*, y por lo tanto, al número de grupos resultantes. Pero este parámetro es dependiente del conjunto de datos mediante estimación y no permite determinar el número de grupos generados. Puede observarse en la figura cómo se han determinado dos grupos para todos los conjuntos de datos.

Debido a los resultados del agrupamiento que se obtienen el algoritmo Mean-Shift no formará parte de la fase de ajuste.

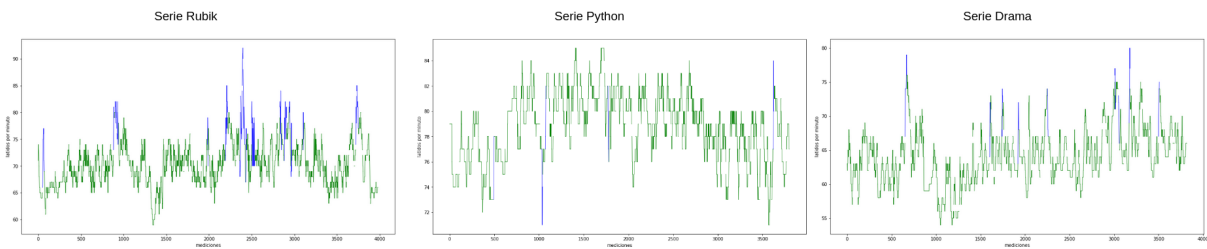


Figura 2.10: Prueba del modelo Mean-Shift.

2.4.7. DBSCAN

El modelo DBSCAN [14] define grupos como regiones continuas de alta densidad. El funcionamiento del algoritmo es el siguiente:

- Para cada elemento del conjunto de datos, el algoritmo cuenta cuántas instancias se sitúan a una distancia pequeña. Esta región se denomina vecindad de la instancia.
- Si una instancia tiene al menos un número mínimo de elementos en su vecindad, se considera una instancia central. Es decir, una instancia central es aquella que está ubicada en una región densa.
- Todos los elementos de la vecindad de una instancia central pertenecen al mismo grupo. Esta vecindad puede incluir otras instancias centrales. Una secuencia de instancias centrales vecinas forma un único grupo.
- Cualquier elemento que no sea una instancia central y no tenga una en su vecindad se considera una anomalía.

DBSCAN es un algoritmo que funciona bien cuando los grupos son lo bastante densos y están separados por regiones de baja densidad. Adicionalmente es un modelo que no requiere indicarle el número de grupos a realizar. En las pruebas realizadas, debido a que los datos no tienen concentraciones segmentadas que generen zonas de alta densidad, no se generaban grupos o se generaban en exceso. Este algoritmo no será seleccionado para realizar la fase de ajuste posterior.

2.4.8. K-Medias con DTW

La comparación directa entre los valores de dos series temporales no permite obtener una medida de similitud adecuada cuando las series temporales están afectadas por alguna deformación o desfase. Una mejor estrategia es realizar un alineamiento temporal previo de las series temporales para posteriormente realizar la comparación mediante una medida de similitud. El algoritmo DTW [38][34] realiza un alineamiento temporal que minimiza la distancia euclídea entre las series. De esta forma

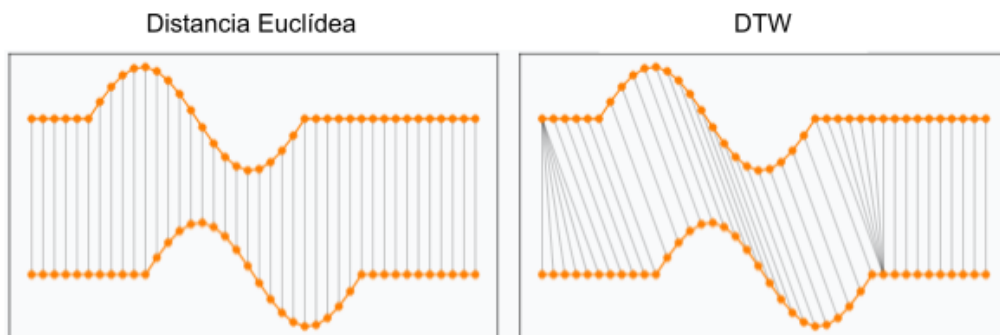


Figura 2.11: En ambos casos la medida de similitud es la suma de las distancias entre los elementos emparejados (conectados por una línea gris).

DTW es capaz de calcular qué transformaciones hay que realizar para alcanzar el mejor alineamiento. En la figura 2.11 puede observarse cómo se produce el alineamiento con el algoritmo DTW.

Las reglas de DTW son las siguientes:

- Cada punto en una serie temporal debe emparejarse con al menos un punto de la otra serie temporal.
- El primer y último índice de cada serie temporal debe emparejarse con su contraparte en la otra serie temporal.
- El mapeo de elementos debe ser un proceso hacia delante que nunca retrocede. Sin embargo, no se necesita avanzar constantemente. Por ejemplo, dos elementos consecutivos de una serie temporal pueden relacionarse con un único elemento de la otra serie temporal.

El algoritmo DTW es el siguiente:

- Entradas: $x_{1:N}$ y $y_{1:N}$
- Matriz de costes: $D \in \mathbb{R}^{N+1 \times M+1}$
- Inicialización:
 - Para $i = 1$ hasta N : $D_{i,0} = \infty$
 - Para $j = 1$ hasta N : $D_{j,0} = \infty$
- Cálculo de la matriz de costes:
 - Para $i = 1$ hasta N :
 - Para $j = 1$ hasta N :

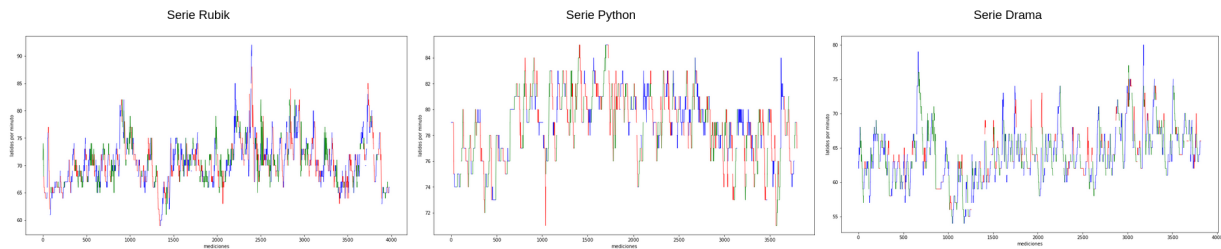


Figura 2.12: Prueba del modelo K-Medias con DTW

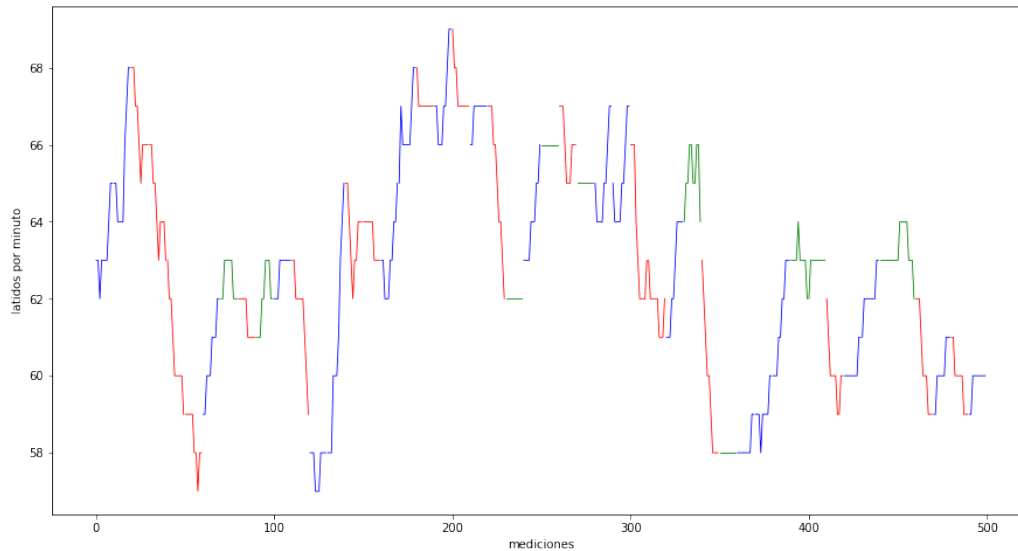


Figura 2.13: Se puede observar en la dirección de los segmentos entre segmentos del mismo color.

$$\diamond D_{i,j} = d(x_i, y_j) + \min \begin{cases} D_{i-1,j-1} & (\text{emparejado}) \\ D_{i-1,j} & (\text{insercción}) \\ D_{i,j-1} & (\text{borrado}) \end{cases}$$

La figura 2.12 muestra un resultado de las pruebas similar al algoritmo original de K-Medias donde encontramos un reparto equilibrado de los segmentos a lo largo del rango de valores de las variables de estudio.

Los resultados obtenidos aplicando DTW a K-Medias son totalmente diferentes a los obtenidos en las pruebas de los anteriores algoritmos. En este caso los grupos se crean en función de la forma del segmento que, debido al reducido tamaño, podemos identificar grupos de segmentos ascendentes y descendentes. Ello conlleva que las gráficas que muestra la figura 2.12 tienen un patrón donde no se produce un asignación por bandas como el algoritmo K-Medias convencional.

Esta asignación debido a la forma que adquiere el segmento es mucho más evidente si se selecciona un fragmento de la serie temporal Rubik (2.13)

La capacidad de agrupar los segmentos por su forma también permite otro tipo de aplicaciones y una interpretación más visual de los resultados.

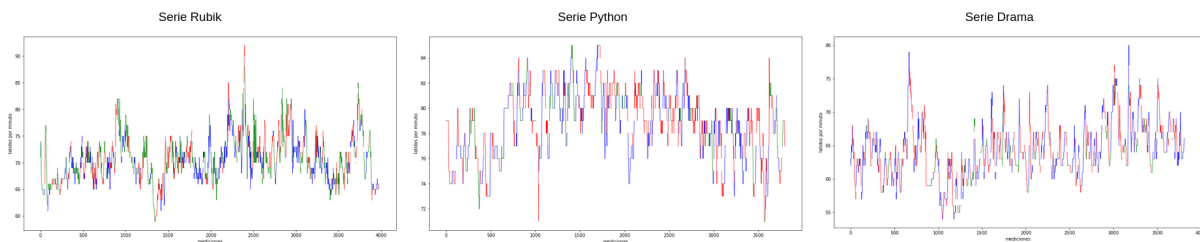


Figura 2.14: Prueba de modelo K-Shape

2.4.9. K-Shape

K-Shape[33][30] está implementado como un proceso iterativo de refinamiento similar a K-Medias y donde redefinen la medida de distancia y el cálculo de los centros de los grupos. Este algoritmo utiliza SBD como medida de distancia basada en la forma. SBD es una alternativa a DTW con mucho menor consumo computacional. SBD realiza un cálculo óptimo de la correlación cruzada que le permite obtener resultados similares a DTW. Este enfoque resulta en una mayor escalabilidad que K-Medias y poder comparar secuencias y calcular centroides sobre series temporales que han sufrido escalado o traslación. El algoritmo de K-Shape sigue los mismos pasos que K-Medias en donde en un primer paso se asignan los elementos a los centroides más cercanos y posteriormente se realiza un paso de refinamiento donde se recalculan los centroides para recoger los cambios realizados en el paso inicial. Este proceso de dos pasos se repite hasta que el algoritmo converge o se alcanzan el número máximo de iteraciones.

Los resultados obtenidos con el algoritmo K-Shape que se muestran en la figura 2.14 muestran un patrón de agrupamiento similar al algoritmo K-Medias con DTW. En este caso no se divide el rango de valores en tramos, ni tampoco se genera un grupo con un gran número de elementos y dos grupos de tamaño muy reducido. En este caso puede apreciarse que el algoritmo agrupa los segmentos ascendentes y descendentes en grupos diferentes. Este resultado se debe a que la medida de similitud utilizada en K-Shape evalúa la forma de la secuencia.

2.5. Ajuste

Tras las pruebas realizadas se seleccionan para la fase de ajuste de hiperparámetros dentro del grupo de algoritmos con cambio de representación: K-medias, BIRCH y agrupamiento por aglomeración. Del grupo de algoritmos que utilizan directamente los datos dinámicos se selecciona K-medias con DTW y K-Shape.

Se han seleccionado dos parámetros para realizar el ajuste de los modelos. El primer parámetro es la longitud de los segmentos en los que se fracciona la serie temporal. Debido a la finalidad que persigue el proyecto, la duración de los segmentos no debería ser muy reducida porque se estaría más cerca de la idea de agrupar medidas de frecuencia cardíaca aisladas, que del concepto de serie

temporal. Pero tampoco los segmentos pueden ser demasiado amplios porque se pretende que el modelo detecte secuencias que reflejen un estado de alteración que no se aleje excesivamente del último momento que incluya cada segmento. Si hay demasiada distancia en tiempo entre el primer y último momento de un segmento, puede que estemos indicando alteraciones que ya no se están produciendo al final del segmento, lo que conlleva que un supuesto mensaje al usuario para que realizara alguna intervención estaría desactualizado. Los posibles valores de los segmentos son 10, 20, 30, 40, 50 y 60 segundos. El segundo parámetro de la fase de ajuste será el hiperparámetro número de grupos que debe generar cada modelo. Con los resultados obtenidos se podrá observar si la opción de generar tres grupos es una opción aceptable o por el contrario se aleja mucho de otros valores más adecuados de número de grupos. Los valores del hiperparámetro número de grupos son 2, 3, 4, y 5.

Para seleccionar los valores que mejor realicen la labor de agrupación es necesario seleccionar una medida de rendimiento que permita evaluar el resultado de las diferentes ejecuciones de los algoritmos. Debido a las restricciones ya mencionadas, esta medida de evaluación no debe requerir información a priori sobre la agrupación de los segmentos de las series temporales. La medida seleccionada es el coeficiente de silueta [37], que es el coeficiente medio de la silueta sobre todas las instancias. El coeficiente de silueta de una instancia está compuesto por dos valores:

- a: la distancia media entre una instancia y todas las otras instancias del mismo grupo
- b: la distancia media entre una instancia y todas las instancias del grupo más cercano

El coeficiente de silueta s para una instancia se define como:

$$s = \frac{b-a}{\max(a,b)}$$

El coeficiente de silueta puede variar entre -1 y +1, lo que le confiere una mayor facilidad de interpretación sobre otras medidas como son el índice Calinski-Harabasz [4] o el índice Davies-Bouldin [10]. Un coeficiente cercano a +1 significa que la instancia está muy metida en su propio grupo y lejos de otros grupos, mientras que un coeficiente cercano a 0 significa que está cerca del límite de un grupo, y por último, un coeficiente cercano a -1 significa que la instancia puede haberse asignado al grupo equivocado.

En las tablas 2.2, 2.3 y 2.4 se detalla el número de grupos con más puntuación en cada una de las diferentes longitudes de segmento para cada uno de los cuatro algoritmos seleccionados

Los rendimientos de los modelos K-Medias y K-Shape no pueden ser medidos con el coeficiente de silueta convencional. Existe un coeficiente de silueta que incorpora DTW implementado en tslearn para establecer la medida de similitud en este tipo de algoritmos.

2.5.1. Stacking

La existencia de algoritmos que agrupan segmentos cuyos valores son de similar magnitud y otros que agrupan mediante la forma del segmento abre la oportunidad a combinarlos para aprovechar

Rubik	K-Medias	BIRCH	Aglomeración	K-Medias DTW	K-Shape
10 seg	2	2	2	2	2
20 seg	2	3	3	2	2
30 seg	2	2	2	2	2
40 seg	2	2	2	2	2
50 seg	2	2	2	2	2
60 seg	2	2	2	2	2

Tabla 2.2: Mejores número de grupos para la serie temporal Rubik.

Python	K-Medias	BIRCH	Aglomeración	K-Medias DTW	K-Shape
10 seg	2	2	2	2	2
20 seg	2	3	3	2	2
30 seg	2	2	2	2	2
40 seg	2	2	2	2	2
50 seg	2	2	2	2	5
60 seg	2	2	2	2	2

Tabla 2.3: Mejores número de grupos para la serie temporal Python.

Drama	K-Medias	BIRCH	Aglomeración	K-Medias DTW	K-Shape
10 seg	2	2	2	2	3
20 seg	2	2	2	2	2
30 seg	2	2	2	2	2
40 seg	2	2	3	2	2
50 seg	2	2	2	2	2
60 seg	2	2	4	2	2

Tabla 2.4: Mejores número de grupos para la serie temporal Drama.

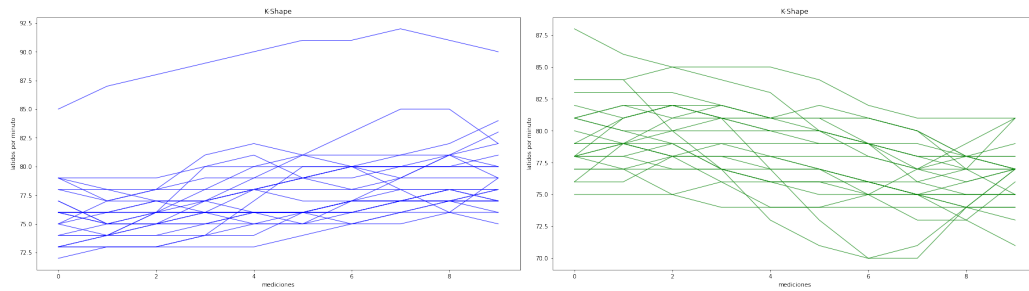


Figura 2.15: Stacking en la serie Rubik. Las imágenes contienen los segmentos del grupo con mayor magnitud en sus valores. La imagen de la izquierda contiene los segmentos que el stack de ambos algoritmos agrupa con un comportamiento generalmente ascendente y a la derecha la imagen del grupo donde recaen los segmentos con una forma descendente.

ambos modos de funcionamiento. Las técnicas de ensamblaje de modelos [3][29] permiten combinar algoritmos, obteniendo mejores resultados mediante la combinación que por sí solos. Con relación al objetivo del proyecto, se podría utilizar inicialmente un modelo que detecta si un segmento se encuentra en la zona de alteración e indicar adicionalmente si lo agrupa como ascendente o descendente. Para ello se crea un esquema de stacking donde se combinan los modelos de K-Medias y K-Medias con DTW. El modelo K-Medias recibirá la serie temporal y los agrupará en tres grupos. Los segmentos del grupo que contiene los valores de mayor magnitud (banda superior) formará el conjunto de entrada del modelo K-Medias con DTW. Éste último modelo realizará una agrupación en dos únicos grupos con el fin de detectar segmentos ascendentes y descendentes. Puede verse el resultado de la combinación de los modelos en la figura 2.15 donde pueden apreciarse en uno de los grupos segmentos ascendentes y en otro descendentes.

La valoración del coeficiente de silueta obtenida para esta combinación de algoritmos es superior a la de cualquiera de los algoritmos por separado.

2.5.2. Adaptación con DTW

El problema que resuelve DTW es conocido un problema clásico a resolver mediante programación dinámica. Este algoritmo puede implementarse en Python de la siguiente forma como se muestra en el cuadro 2.1.

```
from sklearn.metrics import pairwise_distances
from math import sqrt

def distanciaDTW(ts1, ts2):
    DTW={}
    for i in range(len(ts1)):
        DTW[(i, -1)] = np.inf
    for i in range(len(ts2)):
```

```

DTW[(-1, i)] = np.inf
DTW[(-1, -1)] = 0

### Cálculo del óptimo
### Un paso en cada momento de la secuencia
for i in range(len(ts1)):
    for j in range(len(ts2)):
        dist = (ts1[i] - ts2[j])**2
        DTW[(i, j)] = dist + min(DTW[(i-1, j)],
                                  DTW[(i, j-1)],
                                  DTW[(i-1, j-1)])

### Una vez encontrado el camino óptimo
### se devuelve el valor de la distancia
return sqrt(DTW[len(ts1)-1, len(ts2)-1])

```

Con esta función se puede calcular la distancia que hay entre los segmentos en función de su forma. Algunas implementaciones de algoritmos de agrupamiento permiten indicarles mediante parametrización la matriz de afinidad entre los elementos. De esta forma podemos obtener un algoritmo que utiliza la medida DTW o cualquier otra a la hora de ejecutarse. El cuadro 2.2 muestra la adaptación del algoritmo de agrupamiento aglomerativo con la medida DTW anteriormente definida:

```

X = TimeSeriesScalerMeanVariance(mu=0.,
                                  std=1.).fit_transform(df[c].values)

p = pairwise_distances(X.reshape(X.shape[0],
                                  X.shape[1]),
                       metric = distanciaDTW)

ag_dtw = AgglomerativeClustering(n_clusters = N_GRUPOS,
                                  linkage="average",
                                  affinity='precomputed').fit(p)

```

De esta forma se obtienen unos resultados de agrupamiento acordes con una medida que evalúa la forma de los segmentos.

2.6. Despliegue

Para completar la evaluación de la viabilidad del uso de relojes inteligentes en la detección de cambios del estado afectivo mediante algoritmos de aprendizaje automático se desarrolla e implemen-

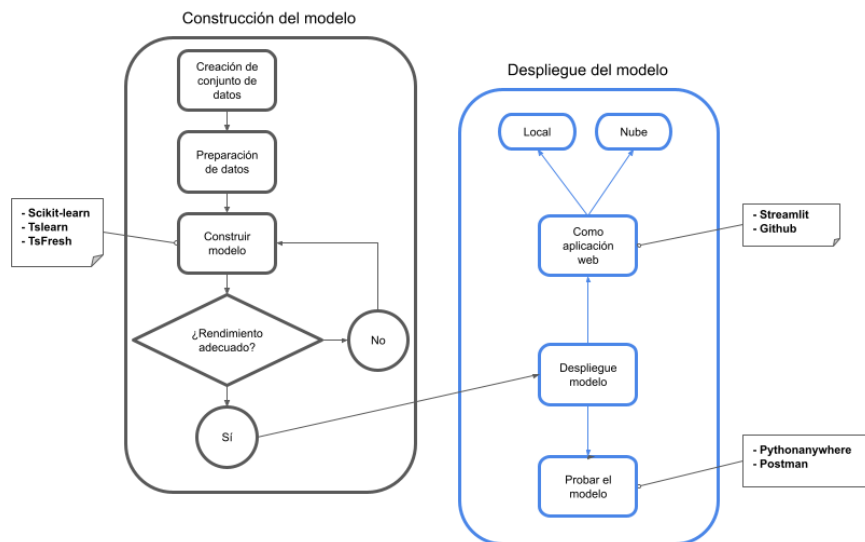


Figura 2.16: Esquema de la construcción y despliegue del modelo

ta una solución en servicios orientados a la producción. Debido a la selección de un algoritmo muy específico no implementado en las plataformas globales de servicios de Aprendizaje Automático y los bajos requerimientos en procesamiento y almacenamiento de datos, el despliegue puede realizarse sin requerir plataformas de servicios de aprendizaje automático. Tradicionalmente este tipo de escenarios se han implementado mediante *notebooks* de Jupyter⁸, API o desarrollando una aplicación web. La opción de notebooks o una API es incompatible con la necesidad de ofrecer una interfaz de usuario con la información actualizada del proceso de agrupamiento en tiempo real. El abordaje tradicional es envolver el modelo en una API y ofrecerla a través de una aplicación web desarrollada con los marcos de trabajo Django⁹ o Flask¹⁰. Estos marcos de trabajo poseen una curva de aprendizaje importante al ser herramientas de desarrollo web generalistas. Un enfoque más ajustado a las necesidades del presente proyecto es utilizar una herramienta de *low-code* para la creación de la interfaz gráfica y la interacción con la API. La plataforma *Streamlit* es una plataforma que persigue cuatro principios de diseño: las aplicaciones web se crean mediante scripts en Python, los componentes gráficos se comportan como variables, uso de mecanismos de caché y despliegue inmediato en la nube. Estos principios acercan los mecanismos *low-code* a los proyectos de ciencia de datos basados en Python. La figura 2.16 contiene un esquema del diseño de la fase de creación y despliegue del modelo seguida en este proyecto, así como las herramientas utilizadas.

Las características básicas que de la solución desplegada son:

- Recolecta los datos generados cada segundo por la pulsera y los representa gráficamente
- Aplica el modelo de K-Medias con cambio de representación. Se selecciona un algoritmo que

⁸Jupyter: <https://jupyter.org/>

⁹Django: <https://www.djangoproject.com/>

¹⁰<https://flask.palletsprojects.com/>

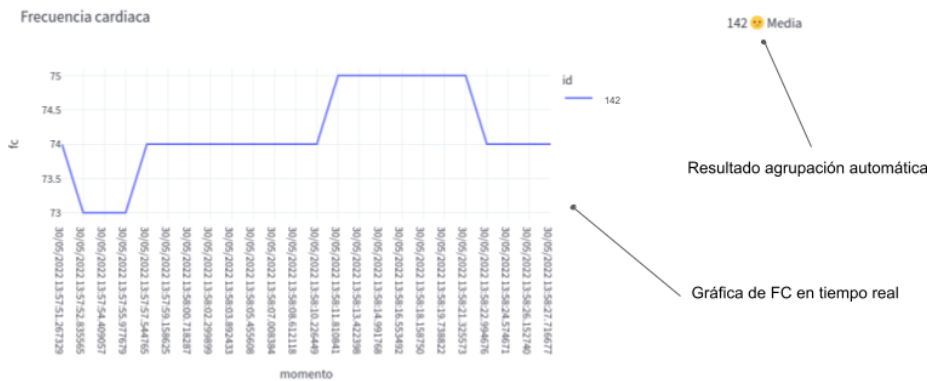


Figura 2.17: Interfaz de la aplicación web

agrupa los segmentos por rango de valores, definiendo el grupo con centroide de mayor magnitud como el nivel de alteración.

- Envía la información del grupo asignado al último segmento recibido a la pulsera
- La pulsera avisa al usuario cuando recibe una indicación de una alteración

En la imagen

La imagen 2.17 muestra la visualización en tiempo real de la frecuencia cardíaca. Puede verse una gráfica que actualiza sus valores cada segundo conforme estos son enviados desde el el reloj inteligente. Una vez que el número de mediciones realizadas supera el mínimo determinado por parámetro, el algoritmo de agrupamiento es ejecutado a intervalos de tiempo fijos para realizar el agrupamiento de los segmentos de la serie temporal. El resultado de la agrupación del último segmento que ha sido procesado por el algoritmo es mostrado mediante un icono y una etiqueta que indica el nivel de alteración. Cuando se alcanza el nivel alto, el reloj inteligente recibe la indicación de que debe avisar al usuario mediante un mensaje tranquilizador de que se encuentra en un nivel alto de FC en relación al período de calibración y las mediciones que ya se encuentran registradas.

Capítulo 3

Resultados

La aplicación desarrollada específicamente para el reloj inteligente ha permitido crear varias series temporales con una resolución temporal de 1 segundo de las variables disponibles y relacionadas con el objetivo del presente proyecto (frecuencia cardiaca y aceleración). La arquitectura de las aplicaciones y la cadena de herramientas propuesta por el fabricante ha permitido el desarrollo y publicación de la aplicación con un contenido nivel de complejidad.

Con la prueba de diferentes modelos de aprendizaje automático no supervisado para agrupamiento se han obtenido tres diferentes resultados:

- Grupos que se reparten el rango de valores de la variable frecuencia cardiaca. Este tipo de resultado muestra gráficamente bandas horizontales de una altura similar. Algoritmos como K-Medias, BIRCH y aglomerativo generan este tipo de patrón. Estos modelos utilizan la distancia euclídea como medida de similitud entre segmentos lo que conlleva una búsqueda de la varianza en los grupos generados. Por lo tanto, tienden a señalar grupos con formas ovaladas (en espacios bidimensionales) alrededor de los centroides. En el caso de espacios unidimensionales con una distribución de elementos continua alrededor de la media los tres centroides se distribuyen a lo largo del rango unidimensional de la variable. La figura 3.1 muestra la distribución de los segmentos de la serie Rubik en los 3 grupos generados por el algoritmo K-Medias en tres bandas bien diferenciadas.
- Agrupación que genera un grupo mayoritario y dos grupos con apenas instancias. Los modelos de agrupamiento espectral y Mean-Shift delimitan la mayor parte de los segmentos alrededor del valor medio de los elementos de la serie temporal en un único grupo. En los otros dos grupos quedan segmentos alejados de esta media formando picos o valles de valores más extremos como puede verse en la figura 3.2.
- Grupos en función de la forma que tienen los segmentos. K-Medias y K-Shape agrupan los segmentos independientemente de la magnitud de los valores que los componen. Ambos modelos utilizan medidas de similitud que evalúan la forma de la secuencia de valores que forma

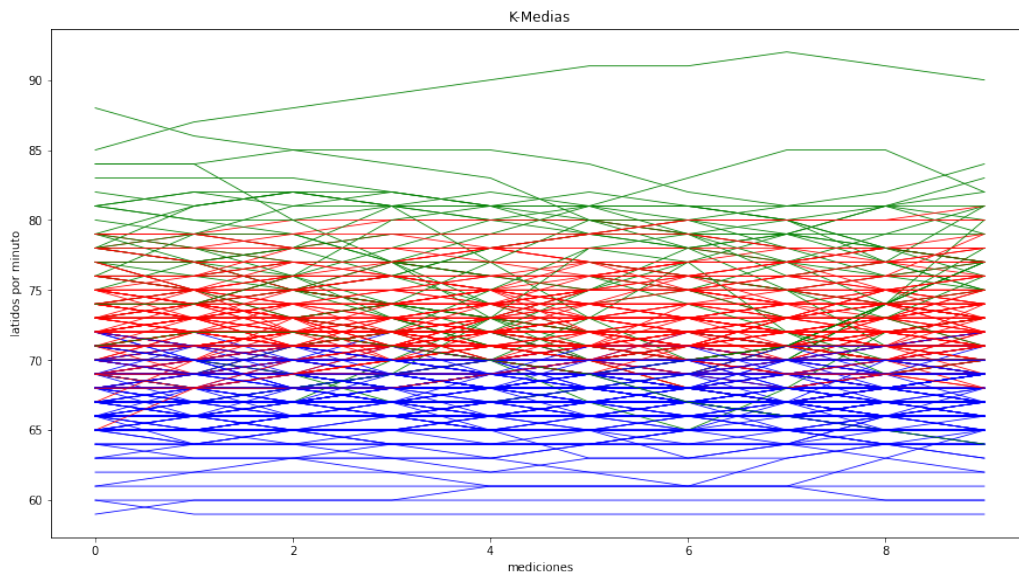


Figura 3.1: Distribución de segmentos en grupos por algoritmo K-Medias.

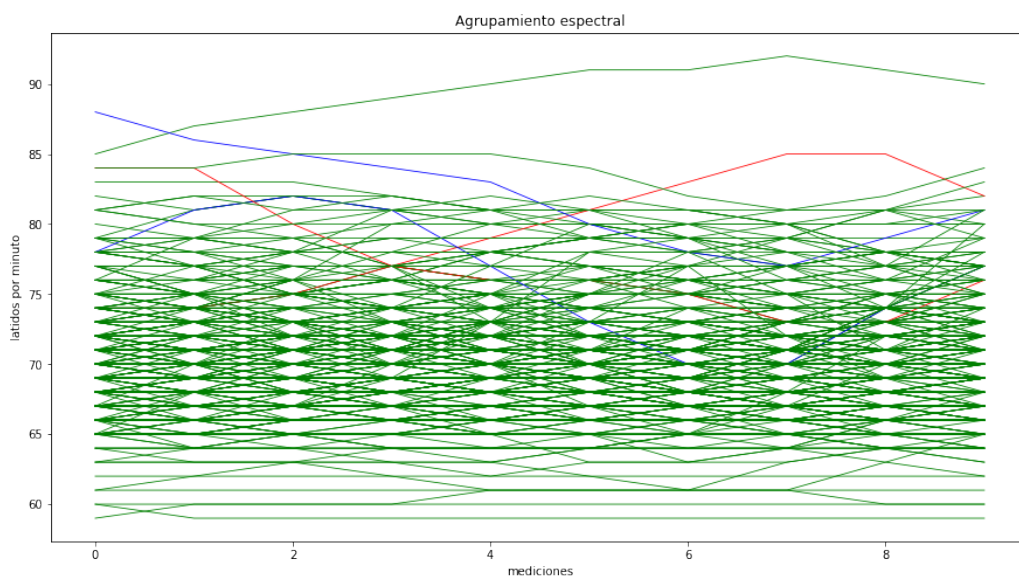


Figura 3.2: Distribución de segmentos en grupos por algoritmo de agrupamiento espectral.

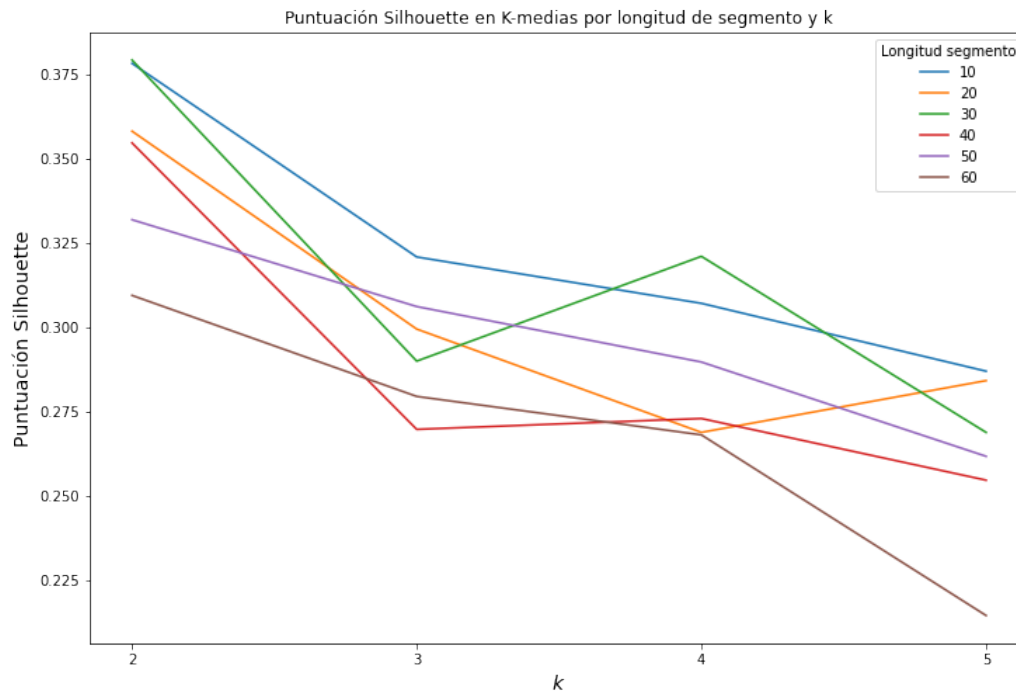


Figura 3.3: Coeficiente silueta en función del parámetro número de grupos en modelo K-Medias para la serie Rubik

cada segmento independientemente de que dichos segmentos estén afectados por deformaciones como el escalado o el desplazamiento. Los modelos agrupan los segmentos en segmentos ascendentes y descendentes.

Dentro del grupo de algoritmos seleccionados que requieren cambio de representación no destaca ninguno por tener un rendimiento muy superior al resto. El algoritmo de agrupamiento por aglomeración presenta los peores resultados en este grupo. La representación gráfica de los grupos generados por ambos algoritmos también es muy similar. Tampoco aparecen rendimientos muy diferentes cuando se utilizan longitudes de segmento de distintos tamaños.

En los resultados del resto de algoritmos no existe una tendencia clara en el rendimiento en cuanto al aumento del tamaño de la ventana que determina un segmento.

La fase de ajuste del hiperparámetro número de grupos muestra que el valor más frecuente dentro de las máximas puntuaciones es 2 grupos. Para la representación y utilización del resultado como retroalimentación al usuario, puede ser más interpretable una valoración de 3 niveles[7] donde se espera que en el período de calibración se encuentre en el nivel bajo de alteración. Se señalarían aquellos segmentos donde son agrupados dentro del grupo con centroide de mayor magnitud. La puntuación de 3 grupos suele ser la segunda mejor puntuación en cada longitud de segmento como muestra el ejemplo de la figura 3.3. Con patrón de agrupamiento que por bandas y definiendo 3 grupos los segmentos identificados como alteración se encontrarán en el tercio superior del rango de valores mostrados por la variable frecuencia cardíaca.

En el grupo de algoritmos que no requieren cambio de representación hay un rendimiento superior del coeficiente de silueta del modelo de K-Medias con DWT sobre K-Shape. Las mejores puntuaciones de K-Medias con DWT se obtienen con segmentos de 10 segundos. En cuanto al número de grupos el valor 2 es el que alcanza mayor puntuación en todas las ejecuciones en el algoritmo de K-medias con DTW. La magnitud de la diferencia entre la puntuación del coeficiente de silueta para los valores de grupos 2 y 3 es mayor en K-Medias con DTW y K-Shape que en el grupo de algoritmos con cambio de representación. Este resultado refuerza la idea de que estos algoritmos que están capturando un patrón de agrupamiento con dos valores: segmentos ascendentes y descendentes.

El coeficiente de silueta no es capaz de valorar positivamente el tipo de agrupamiento basado en la forma. Para valorar el rendimiento de algoritmos que miden similitudes en la forma de las series temporales es necesario disponer de implementaciones de medidas de rendimiento que puedan valorar este aspecto. En este caso se utiliza una implementación del coeficiente de silueta que utiliza la medida de DWT.

La selección de la medida de similitud tiene un gran impacto en los resultados de los algoritmos, sobre todo en el ámbito de las series temporales[42]. Se obtienen resultados positivos con medidas que incorporan la evaluación de la forma de la secuencia de valores que componen la serie. También es posible adaptar otros algoritmos que admitan la parametrización de la matriz de afinidad con este tipo de medidas.

Se definirá por tanto como configuración predeterminada para el despliegue la utilización del algoritmo K-Medias agrupando en 3 grupos con una duración de segmentos de 10 segundos.

El desarrollo de la interfaz web para la ejecución en tiempo real requirió una curva de aprendizaje contenida debido a que la solución de low-code seleccionada oculta los detalles tecnológicos necesarios para el desarrollo de este tipo de interfaces.

Capítulo 4

Conclusiones

La principal de las conclusiones alcanzadas en este trabajo es que es posible automatizar el proceso completo de detección de cambios afectivos en tiempo real a partir de los datos que capta un reloj inteligente. La incorporación de algoritmos de aprendizaje automático para realizar la labor de detección sin información previa puede ser una solución alternativa al uso de datos etiquetados previamente mediante cuestionarios de autoevaluación. Esto conlleva la posibilidad de crear un flujo de trabajo más simple y adaptado a las particularidades de cada individuo.

El resultado más relevante es que el uso de algoritmos de aprendizaje no supervisado es una opción adecuada para detectar patrones en series temporales. El empleo de modelos que actúan directamente sobre los datos de las series temporales sin requerir un cambio de representación aporta simplicidad a la vez que unos rendimientos óptimos al modelo final. Adicionalmente estos algoritmos permiten orientar más adecuadamente la tarea de agrupamiento en relación con la forma de la serie temporal.

Es importante señalar que ante la falta de estándares o interfaces que provean buena calidad de la información, el desarrollo de una aplicación específica para el reloj inteligente es la vía más adecuada para garantizar un acceso lo más completo posible a los datos captados por los sensores.

Cabe también destacar que el empleo de herramientas de low-code en la fase de despliegue de un modelo aporta un abordaje eficiente para la construcción de la pieza de software que pondrá en relación la API del modelo con el usuario. Estas herramientas permiten crear una interfaz para el usuario que le permita visualizar el resultado del modelo en producción, así como interactuar con sus parámetros.

Sin embargo, en este trabajo no se han podido incorporar datos de variables muy relevantes para la detección de cambios en estados afectivos. Los sensores de los que disponen los relojes inteligentes analizados en este trabajo disponen de sensores que registran la actividad electrodermal y el ritmo cardiaco, pero no son accesibles a través de otra vía que no sea la visualización en la aplicación del teléfono inteligente al que están conectados. Los mecanismos y políticas restrictivas de acceso a los datos que cada fabricante implanta hacen necesario un estudio de las capacidades para extraer datos de los modelos más relevantes del mercado. Derivada de esta incapacidad de acceder a más

variables por el diseño de la interfaz del reloj, no ha sido posible explorar los resultados que se habrían generado al utilizar otras variables significativas en la detección de estados afectivos. La utilización de un mayor uso de variables conlleva realizar un estudio de qué cambios de representación serían los más adecuados para la tarea de detección. Este trabajo ha evidenciado que la tecnología actual permite a día de hoy medir una gran diversidad de variables muy útiles que van más allá del ámbito clínico. En el futuro irán apareciendo nuevos modelos de relojes inteligentes que tendrán cada vez más capacidad de medir variables, así como autonomía frente al teléfono inteligente con el que se sincroniza. Estos dispositivos deberán mejorar sus mecanismos de accesibilidad a los datos capturados por los sensores para permitir la creación de un ecosistema rico en soluciones y aplicaciones que extraigan valor de los datos mediante técnicas avanzadas de aprendizaje automático y profundo.

Bibliografía

- [1] Aghabozorgi, S., Shirkhorshidi, S., and Wah, T. (2015). Time-series clustering—a decade review. *Information Systems*, 53:16–38.
- [2] Andrew, N., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14.
- [3] Burkov, A. (2019). *The Hundred Page Machine Learning Book*. Andriy Burkov.
- [4] Caliński, T. and J., H. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27.
- [5] Can, Yekta Said, e. a. (2019). Continuous stress detection using wearable sensors in real life: Algorithmic programming contest case study. *Sensors*, 19(8):1849.
- [6] Can, Y., Chalabianloo, N., Ekiz, D., Fernandez-Alvarez, J., Repetto, C., Riva, G., Iles-Smith, H., and Ersoy, C. (2020a). Real-life stress level monitoring using smart bands in the light of contextual information. *IEEE Sensors Journal*.
- [7] Can, Y. S., Chalabianloo, N., Ekiz, D., Fernandez-Alvarez, J., Riva, G., and Ersoy, C. (2020b). Personal stress-level clustering and decision-level smoothing to enhance the performance of ambulatory stress detection with smartwatches. *IEEE Access* 8, 8:38146–38163.
- [8] Chudy, N. S. (2017). *Testing of Wrist-Worn-Fitness-Tracking Devices During Cognitive Stress: A Validation Study*. PhD thesis, University of Central Florida University of Central.
- [9] Cohen, S., Kamarck, T., and Mermelstein, R. (1983). A global measure of perceived stress. *Health Social Behaviour*, 24(4):385.
- [10] Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):224–227.
- [11] Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., and Keogh, E. (2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. *PVLDB*, 1:1542–1552.

- [12] Dorin, C. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619.
- [13] Egilmez, B. (2017). Ustress: Understanding college student subjective stress using wrist-based passive sensing. *EEE International Conference on Pervasive Computing and Communications Workshops*, pages 673–678.
- [14] Ester, M., Kriegel, H. P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.
- [15] F., G., A., M., G., T., B., A., and A., G. (2013). Towards a mobile galvanic skin response measurement system for mentally disordered patients. *Proceedings of the 8th International Conference on Body Area Networks*, pages 432–435.
- [16] Factors, F. (2022). Demand for global smartwatch market size - share worth usd 97.52 billion by 2028, exhibit a cagr of 21.5value, analysis and forecast report 2022-2028 | opportunities and challenges | facts - factors.
- [17] Fauzi, M. A. and Yang, B. (2021). Continuous stress detection of hospital staff using smartwatch sensors and classifier ensemble. *pHealth 2021*, pages 245–250.
- [18] Gevirtz, R. (2013). The promise of heart rate variability biofeedback: evidence-based applications. *Biofeedback*, 41(3).
- [19] Greco, A., Valenza, G., Lanata, A., Rota, G., and Scilingo, E. P. (2014). Electrodermal activity in bipolar patients during affective elicitation. *IEEE Journal of Biomedical and Health Informatics*, 18(6):1865–1873.
- [20] Guddorp, D. (2017). *Stress response coherence: Does self-perceived stress and its interaction with physiological measures depend on the kind of stressor?* PhD thesis, University of Twente. Enschede.
- [21] Géron, A. (2020). *Aprende Machine Learning con Scikit-Learn, Keras y TensorFlow*.
- [22] Hao, T., Walter, K. N., Ball, M. J., Chang, H. Y., Sun, S., and Zhu, X. (2017). Stresshacker: towards practical stress monitoring in the wild with smartwatches. *AMIA Annual Symposium Proceedings*, 2017:830.
- [23] Hart, S. G. and Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183.
- [24] Hänsel, K., Alomainy, A., and Haddadi, H. (2016). Large scale mood and stress self-assessments on a smartwatch. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1180–1184.

- [25] J, F. B. and Dueck, D. (2007). Clustering by passing messages between data points. *science*, 315(5814):972–976.
- [26] Katarya, R. and Maan, S. (2020). Stress detection using smartwatches with machine learning: a survey. *International Conference on Electronics and Sustainable Communication Systems*, pages 306–310.
- [27] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–287.
- [28] Mahnoosh, S., McDonald, A. D., and Sasangohar, F. (2022). Posttraumatic stress disorder hyperarousal event detection using smartwatch physiological and activity data. *Plos one*, 17(5):e0267749.
- [29] Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018). The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808.
- [30] Martinez-Tejada, I., Riedel, C. S., Juhler, M., Andresen, M., and Wilhjelm, J. E. (2022). k-shape clustering for extracting macro-patterns in intracranial pressure signals. *Fluids Barriers CNS*, 19(1):1–13.
- [31] McCraty, R. and Shaffer, F. (2015). Heart rate variability: new perspectives on physiological mechanisms, assessment of self-regulatory capacity, and health risk. *Global advances in health and medicine*, 4(1):46–61.
- [32] Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The computer journal*, 26(4):354–359.
- [33] Paparrizos, J. and Gravano, L. (2015). k-shape: Efficient and accurate clustering of time series. In *ACM SIGMOD international conference on management of data*, pages 1855–1870.
- [34] Petitjean, F., Ketterlin, A., and Gançarsk, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern recognition*, 44(3):687–693.
- [35] Picard, R. W. (2016). Automating the recognition of stress and emotion: From lab to real-world impact. *IEEE MultiMedia*.
- [36] Rodríguez-Núñez, I., Rodríguez-Romero, N., Álvarez, A., Zambrano, L., Luciano da Veiga, G., and Romero, F. (2022). Variabilidad del ritmo cardíaco en pediatría: aspectos metodológicos y aplicaciones clínicas. *Archivos de cardiología de Mexico*, 92(2):242–252.
- [37] Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, (20):53.

- [38] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49.
- [39] Salai, M., Vassányi, I., and Kósa, I. (2016). Stress detection using low cost heart rate sensors. *Journal of Healthcare Engineering*, 2016.
- [40] Santos, O. C., Boticario, J. G., Cabestrero, R., and Quirós, P. (2013). Extending collaborative learning modeling with emotional information. In *Proceedings of the CEUR*.
- [41] Santos, O. C., Boticario, J. G., Cabestrero, R., and Quirós, P. (2021). El marco lógico colaborativo como herramienta inteligente para la innovación docente aprovechando la computación afectiva.
- [42] Serra, J. and Arcos, J. L. (2014). An empirical evaluation of similarity measures for time series classification. *Knowledge-Based Systems*, 67:305–314.
- [43] Shaffer, F., Meehan, Z. M., and Zerr, C. L. (2020). A critical review of ultra-short-term heart rate variability norms research. *Frontiers in neuroscience*, 14:594880.
- [44] Siirtola, P. (2019). Continuous stress detection using the sensors of commercial smartwatch. *Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pages 1198–1201.
- [45] Umair, Muhammad, e. a. (2021). Hrv and stress: a mixed-methods approach for comparison of wearable heart rate sensors for biofeedback. *IEEE Access*, 9:14005–14024.
- [46] van Lier, H. G., Pieterse, M. E., Garde, A., Postel, M. G., de Haan, H. A., Vollenbroek-Hutten, M. M. R., Schraagen, J. M., and Noordzij, M. L. (2020). A standardized validity assessment protocol for physiological signals from wearable technology: Methodological underpinnings and an application to the e4 biosensor. *Behavior Research Methods*, 52(2):607–629.
- [47] Villarejo, M. V., Zapirain, B. G., and Zorrilla*, A. M. (2012). A stress sensor based on galvanic skin response (gsr) controlled by zigbee. *Sensors (Basel)*, 12(5):6075–6101.
- [48] Ward, J. and Joe, H. (1963). Hierarchical grouping to optimize an objective function. *American statistical association*, 58(301):263–244.
- [49] Warren, L. T. (2005). Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874.
- [50] Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, (16):645–678.
- [51] Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114.