



**UNIVERSIDAD  
DE GRANADA**

TRABAJO FIN DE GRADO  
INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

# **Supervisión del Rendimiento y la Recuperación de Deportistas con Dispositivos Wearables**

---

Caso de estudio: deportes de contacto.

**Autor**

Marcos Tendero Carmona

**Directores**

Juan Antonio Holgado Terriza



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

—  
Granada, Noviembre de 2024









**UNIVERSIDAD  
DE GRANADA**

# **Supervisión del Rendimiento y la Recuperación de Deportistas con Dispositivos Wearables.**

---

Caso de estudio: deportes de contacto.

**Autor**

Marcos Tendero Carmona

**Directores**

Juan Antonio Holgado Terriza



# **Supervisión del Rendimiento y la Recuperación de Deportistas con Dispositivos Wearables. Caso de estudio: deportes de contacto**

Marcos Tendero Carmona

**Palabras clave:** dispositivo wearable, reloj inteligente, red personal, VO<sub>2</sub> máximo, Variabilidad de la Frecuencia Cardíaca (HRV), DataFrame (DF).

## **Resumen**

En una sociedad digital en perpetuo crecimiento, la búsqueda de soluciones tecnológicas que optimicen la vida de las personas de forma cómoda y no invasiva ha fomentado el desarrollo de dispositivos IoT portátiles, conocidos como wearables, aplicados a aspectos de la vida cotidiana como la salud y el deporte.

Este proyecto tuvo como objetivo desarrollar un sistema de supervisión de la recuperación y el rendimiento de deportistas de contacto mediante dispositivos wearables. Este sistema se ha estructurado en tres fases: la búsqueda y configuración de una red personal de dispositivos adecuada a los requisitos de los deportes de contacto; el diseño y despliegue de un sistema de monitorización que permita capturar los datos necesarios para evaluar el rendimiento y la recuperación de los deportistas; y la implementación de un sistema de análisis de los datos capturados, que evalúa la recuperación y el rendimiento de los deportistas mediante un algoritmo de calificación basado en una interpretación de la literatura revisada en ciencias de la salud y el deporte.

Aunque aún falta una validación rigurosa del sistema por parte de especialistas en ciencias de la salud y el deporte con un conjunto de voluntarios mucho más amplio en el estudio, el desarrollo del proyecto ha demostrado la capacidad de los sistemas de monitorización con dispositivos wearables para mejorar la experiencia de los usuarios en la práctica deportiva.



# **Supervisión del Rendimiento y la Recuperación de Deportistas con Dispositivos Wearables. Caso de estudio: deportes de contacto**

Marcos Tendero Carmona

**Keywords:** wearable device, smartwatch, personal network, VO<sub>2</sub> max, Heart Rate Variability (HRV), DataFrame (DF).

## **Abstract**

In a perpetually growing digital society, the search for technological solutions that optimize people's lives in a comfortable and non-invasive manner has driven the development of portable IoT devices, known as wearables, applied to everyday aspects such as health and sports.

This project aimed to develop a system for monitoring the recovery and performance of combat athletes using wearable devices. The system was structured into three phases: the identification and setup of a personal network of wearable devices tailored to the needs of contact sports; the design and deployment of a monitoring system that captures the required data for evaluating athletes' performance and recovery; and the implementation of a data processing and analysis system that assesses athletes' recovery and performance using a scoring algorithm based on a review of health and sports science literature.

Although a rigorous validation of the monitoring system by health and sports science specialists, along with a much larger group of volunteers in the study, is still needed, the development of this project has demonstrated the potential of wearable monitoring systems to enhance user experience in sports practice.



---

Yo, **Marcos Tendero Carmona**, alumno de la titulación Grado en Ingeniería de Tecnologías de Telecomunicación de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77143845M, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Marcos Tendero Carmona

Granada a 12 de Noviembre de 2024.



---

**D. Juan Antonio Holgado Terriza**, Profesor del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada.

**Informan:**

Que el presente trabajo, titulado *Supervisión del Rendimiento y la Recuperación de Deportistas con Dispositivos Wearables. Caso de estudio: deportes de contacto*, ha sido realizado bajo su supervisión por **Marcos Tendero Carmona**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 12 de Noviembre de 2024.

**Los directores:**

**Juan Antonio Holgado Terriza**



# Agradecimientos

En primer lugar, quería agradecer a los compañeros y profesores que me han acompañado durante mis años en la Escuela Técnica Superior de Ingeniería Informática y de Telecomunicación, pero sobretodo a los amigos que me deja esta etapa universitaria, que han hecho más corto un maravilloso camino que está llegando a su fin.

Por otro lado, quería dar las gracias a mi tutor, D. Juan Antonio Holgado Terriza, por su trabajo y tiempo empleados en este Proyecto y por el aprendizaje que me llevo de nuestras reuniones y charlas.

Como último y más importante, gracias papá, mamá, Hugo y Opi, por confiar en mí y apoyarme siempre, guiándome en los peores momentos.

Muchas gracias a todos,

Marcos.



# Índice general

<b>1. Introducción</b>	<b>23</b>
1.1. Motivación . . . . .	23
1.2. Objetivos . . . . .	24
1.3. Planificación . . . . .	24
1.3.1. Fase 1: Planteamiento de los objetivos . . . . .	25
1.3.2. Fase 2: Documentación teórica del Proyecto . . . . .	25
1.3.3. Fase 3: Desarrollo experimental y técnico . . . . .	25
1.3.4. Fase 4: Captura de los datos de estudio . . . . .	26
1.3.5. Fase 5: Análisis de los datos capturados . . . . .	26
1.3.6. Fase 6: Redacción del informe . . . . .	26
1.4. Componentes utilizados . . . . .	28
1.4.1. Hardware . . . . .	28
1.4.2. Software . . . . .	28
1.4.3. Recursos humanos . . . . .	28
1.5. Presupuesto . . . . .	28
1.5.1. Presupuesto de la infraestructura de Hardware . . . . .	28
1.5.2. Presupuesto de la infraestructura de Software . . . . .	28
1.5.3. Presupuesto recursos humanos . . . . .	29
1.5.4. Presupuesto Total . . . . .	29
1.6. Estructura del Proyecto . . . . .	29
<b>2. Estado del Arte</b>	<b>31</b>
2.1. Dispositivo Wearable . . . . .	31
2.1.1. Características del Dispositivo Wearable . . . . .	31
2.1.2. Arquitectura de un sistema de monitorización con dispositivos wearables .	32
2.1.3. Tipos de wearables . . . . .	33
2.1.4. Análisis del mercado de dispositivos wearables. Diseño de una red personal	34
2.2. Caracterización del entrenamiento de deportes de contacto . . . . .	35
2.2.1. Estructura temporal y caracterización física . . . . .	35
2.2.2. Rutina de boxeo de un gimnasio amateur . . . . .	36
2.2.3. Parámetros de interés medibles con wearables . . . . .	38
<b>3. Materiales y métodos</b>	<b>49</b>
3.1. Hardware empleado . . . . .	49
3.1.1. Reloj inteligente Garmin Venu 2 . . . . .	49
3.1.2. Sensor de frecuencia cardíaca Garmin HRM-Pro Plus. . . . .	53
3.1.3. Arquitectura de la red personal . . . . .	54
3.1.4. Sincronización de ambos dispositivos . . . . .	56
3.2. Software empleado . . . . .	56
3.2.1. Anaconda Navigator . . . . .	56
3.2.2. Aplicaciones de Garmin utilizadas . . . . .	58

## ÍNDICE GENERAL

---

3.3.	Algoritmo de calificación de la recuperación y el rendimiento deportivo . . . . .	60
3.3.1.	Calificación de la recuperación . . . . .	61
3.3.2.	Calificación del rendimiento deportivo . . . . .	64
3.3.3.	Evaluación de la eficiencia del algoritmo . . . . .	67
3.4.	Encuesta SUS para la valoración de la usabilidad . . . . .	68
<b>4.</b>	<b>Desarrollo Experimental y Técnico</b>	<b>71</b>
4.1.	Desarrollo Experimental . . . . .	71
4.1.1.	Procedimiento a seguir por el usuario . . . . .	71
4.1.2.	Configuración y puesta a punto de los dispositivos . . . . .	75
4.1.3.	Usuarios voluntarios del estudio . . . . .	77
4.2.	Desarrollo Técnico . . . . .	78
4.2.1.	Desarrollo fallido de una aplicación Garmin Connect IQ . . . . .	79
4.2.2.	Estructura de la implementación del sistema de análisis de la recuperación y del rendimiento deportivo . . . . .	80
4.2.3.	Preparación de los datos para el análisis . . . . .	80
4.2.4.	Análisis segmentado de los datos . . . . .	82
4.2.5.	Aplicación del algoritmo de calificación de la recuperación y el rendimiento	95
<b>5.</b>	<b>Análisis de Resultados</b>	<b>107</b>
5.1.	Análisis del rendimiento y la recuperación en sus parámetros individuales. . . . .	107
5.1.1.	Variables Basales . . . . .	107
5.1.2.	Frecuencia Cardíaca . . . . .	108
5.1.3.	Variabilidad de la Frecuencia Cardíaca . . . . .	113
5.1.4.	Batería Corporal . . . . .	115
5.1.5.	Balance Calórico . . . . .	117
5.1.6.	Frecuencia Respiratoria . . . . .	118
5.1.7.	Nivel de Estrés . . . . .	119
5.1.8.	Calidad del Sueño . . . . .	120
5.1.9.	VO <sub>2</sub> Máximo tras la Semana de Entrenamiento . . . . .	122
5.2.	Análisis global. Resultados del algoritmo propuesto . . . . .	122
5.2.1.	Resultados de la implementación del algoritmo en Python . . . . .	122
5.2.2.	Ánalisis de los resultados de la recuperación y el rendimiento mediante el algoritmo propuesto . . . . .	124
5.2.3.	Evaluación de la eficiencia del algoritmo propuesto . . . . .	125
5.3.	Análisis del rendimiento de la red personal. . . . .	127
5.3.1.	Autonomía . . . . .	127
5.3.2.	Valoración de la usabilidad mediante encuesta SUS . . . . .	128
<b>6.</b>	<b>Conclusiones y líneas futuras</b>	<b>129</b>
6.1.	Análisis de los Objetivos Completados . . . . .	129
6.2.	Líneas Futuras . . . . .	131
6.3.	Conclusiones personales . . . . .	131
<b>Bibliografía</b>		<b>137</b>
<b>A. Implementación del depurado de datos en Python</b>		<b>139</b>
A.0.1.	Módulos y librerías empleados . . . . .	139
<b>B. Código complementario para la implementación del análisis de datos en Python</b>		<b>145</b>
B.0.1.	Gestión de directorios . . . . .	145
B.0.2.	Funciones de análisis de parámetros . . . . .	145

## ÍNDICE GENERAL

---

<b>C. Código complementario para la implementación del algoritmo de calificación en Python</b>	<b>161</b>
C.0.1. Funciones de evaluación de los parámetros de evaluación . . . . .	161
C.0.2. Funciones de calificación . . . . .	169
<b>D. Herramientas software utilizadas</b>	<b>175</b>
<b>Glosario. Listado de acrónimos.</b>	<b>176</b>



# Índice de figuras

2.1. Arquitectura de un sistema de monitorización con wearables. Elaboración propia.	33
3.1. Smartwatch Garmin Venu 2 . . . . .	49
3.2. Esquema de botones del Garmin Venu 2 . . . . .	51
3.3. Sensor de frecuencia cardíaca Garmin HRM-Pro Plus. . . . .	54
3.4. Capa de captación. Arquitectura de la red personal. Elaboración propia. . . . .	55
3.5. Launcher de Jupyter Notebooks en Anaconda Navigator. . . . .	57
3.6. Display de la prueba Resumen de Salud en un reloj Garmin. . . . .	59
3.7. Pantalla inicial y display de la actividad Barra en nuestro Venu 2 . . . . .	60
4.1. Configuración del modo de grabación del Venu 2 desde Garmin Connect . . . . .	76
4.2. Protector de pantalla de silicona y vendaje de boxeo. . . . .	76
4.3. Disposición de los dispositivos wearables para el entrenamiento. . . . .	77
4.4. Disposición de los dispositivos wearables para el entrenamiento. . . . .	77
4.5. Boceto inicial de la aplicación Connect IQ fallida . . . . .	79
4.6. Diagrama de flujo de la implementación del sistema de monitorización diseñado .	80
5.1. Frecuencia cardíaca durante el entrenamiento 1, usuario 1 . . . . .	109
5.2. Evolución semanal de la frecuencia cardíaca media, máxima y mínima, usuario 1	109
5.3. Evolución semanal de la frecuencia cardíaca media, máxima y mínima, usuario 2	109
5.4. Picos y valles de frecuencia cardíaca durante el entrenamiento 1, usuario 1 . . .	111
5.5. Evolución semanal de la media de picos y valles de frecuencia cardíaca, usuario 1	112
5.6. Evolución semanal de la media de picos y valles de frecuencia cardíaca, usuario 2	113
5.7. Evolución del HRV un día de entrenamiento, usuario 1 . . . . .	113
5.8. Evolución del HRV un día de descanso, usuario 1 . . . . .	113
5.9. Evolución temporal del HRV durante la semana de entrenamiento, usuario 1 . .	114
5.10. Evolución temporal del HRV durante la semana de entrenamiento, usuario 2 . .	114
5.11. Datos corruptos de estrés diario y batería corporal, usuario 1 . . . . .	116
5.12. Nivel de estrés y batería corporal a lo largo del día, usuario 2 . . . . .	116
5.13. Consumo calórico diario por tipo de actividad, usuario 1 . . . . .	117
5.14. Consumo calórico diario por tipo de actividad, usuario 2 . . . . .	117
5.15. Frecuencia respiratoria media pre, durante y post entrenamiento a lo largo de la semana de estudio, usuario 1 . . . . .	118
5.16. Frecuencia respiratoria media pre, durante y post entrenamiento a lo largo de la semana de estudio, usuario 2 . . . . .	118
5.17. Balance de estrés pre/post entrenamiento, usuario 1 . . . . .	120
5.18. Balance de estrés pre/post entrenamiento, usuario 2 . . . . .	120
5.19. Zonas de sueño ligero, profundo y REM a lo largo de un periodo de sueño (FAIR), usuario 1 . . . . .	121
5.20. Zonas de sueño ligero, profundo y REM a lo largo de un periodo de sueño (EX- CELENT), usuario 1 . . . . .	121

## ÍNDICE DE FIGURAS

5.21. Calificaciones de la calidad del sueño a lo largo de la semana de entrenamientos, usuario 1 . . . . .	121
5.22. Calificaciones de la calidad del sueño a lo largo de la semana de entrenamientos, usuario 2 . . . . .	122
5.23. Tabla de calificaciones de recuperación y rendimiento, usuario 1 . . . . .	123
5.24. Tabla de calificaciones de recuperación y rendimiento, usuario 2 . . . . .	123
5.25. Tabla de calificaciones de recuperación y rendimiento, usuario 2 . . . . .	123
5.26. Desplegables de insights del Lunes plegados . . . . .	124
5.27. Desplegables de insights del Lunes desplegados . . . . .	124
5.28. Desplegables de insights un día sin entrenamiento . . . . .	124
5.29. Insights del entrenamiento del jueves, usuario 2 . . . . .	125

# Índice de listings

4.1. Carga de archivos con Pandas . . . . .	83
4.2. Carga de archivos con json load . . . . .	83
4.3. Bucle de extraccion de datos de frecuencia cardiaca . . . . .	85
4.4. Diccionarios para el filtrado manual de picos y valles por su índice . . . . .	86
4.5. Bucle de extraccion de datos de picos de frecuencia cardiaca . . . . .	87
4.6. Bucle de extraccion y representacion de datos de HRV diario . . . . .	88
4.7. Bucle de extracción de datos de estrés y respiración pre/post entrenamiento . . . . .	90
4.8. Diccionario para definir el periodo de entrenamiento para los archivos de estrés y Body Battery . . . . .	91
4.9. Distinción pre/post entrenamiento en el bucle de extraccion y representacion de datos de estres y Body Battery . . . . .	92
4.10. Diccionario con el gasto calórico de cada entrenamiento . . . . .	93
4.11. Funciones de procesado para las graficas de sueño . . . . .	94
4.12. Función de evaluación de los parámetros de FC . . . . .	96
4.13. Función de calificación de la recuperación . . . . .	99
4.14. Visualización de la tabla de calificaciones y los desplegables de insights . . . . .	104
A.1. Módulos y librerías importados para el depurado de datos . . . . .	139
A.2. Directorios para el depurado de datos . . . . .	139
A.3. Funciones para el depurado de datos . . . . .	140
A.4. Bucle principal para el depurado de datos . . . . .	143
B.1. Funciones de gestión de directorios . . . . .	145
B.2. Variables iniciales para frecuencia cardiaca . . . . .	145
B.3. Representación de la frecuencia cardíaca durante el entrenamiento . . . . .	146
B.4. DF y graficas de evolucion frecuencia cardiaca . . . . .	146
B.5. Variables iniciales para picos de frecuencia cardiaca . . . . .	147
B.6. Representación de picos y valles en la gráfica de frecuencia cardíaca durante el entrenamiento . . . . .	148
B.7. DF de picos de frecuencia cardiaca . . . . .	148
B.8. Grafica de evolucion de picos de frecuencia cardíaca . . . . .	149
B.9. Variables iniciales para HRV . . . . .	149
B.10. DF de HRV . . . . .	150
B.11. Representación de la evolución diaria del HRV . . . . .	150
B.12. Representación de la comparación del HRV diario a lo largo de la semana . . . . .	150
B.13. Variables iniciales para FR y estres pre/post entrenamiento . . . . .	151
B.14. Bucle de extracción de datos de estrés y respiración pre/post entrenamiento . . . . .	151
B.15. Representación de la FR y el estrés pre y post entrenamiento . . . . .	153
B.16. Variables iniciales para estrés y Body Battery . . . . .	154
B.17. Primera parte del bucle de extracción y representacion de datos de estres y Body Battery. . . . .	154
B.18. Representación de la batería corporal junto al estrés diario . . . . .	155
B.19. DF de estrés y Body Battery . . . . .	156
B.20. Variables iniciales para uso calorico . . . . .	156

## ÍNDICE DE LISTINGS

---

B.21.Bucle de extraccion de datos de gasto calorico . . . . .	156
B.22.DF de gasto calorico . . . . .	157
B.23.Representación del gasto calórico diario . . . . .	157
B.24.Variables iniciales para el análisis de la calidad del sueño . . . . .	158
B.25.Bucle de extraccion y representacion de datos de sueño . . . . .	158
B.26.Construccion de tabla de calificaciones de sueño . . . . .	159
C.1. 'Diccionarios de Insights de recomendaciones para los distintos parámetros . . . . .	161
C.2. 'Función de evaluación de VO2 Máximo' . . . . .	163
C.3. 'Función de evaluación del HRV' . . . . .	164
C.4. 'Función de evaluación del sueño' . . . . .	165
C.5. 'Función de evaluación del balance de respiración pre/post entrenamiento' . . . . .	166
C.6. 'Función de evaluación de estrés y body battery' . . . . .	167
C.7. Diccionarios con pesos para la calificación de la recuperación y el rendimiento . . . . .	169
C.8. Función de calificación del rendimiento . . . . .	170

# Índice de cuadros

1.1.	Diagrama de Gantt: Octubre 2023 al 25 de marzo de 2024 . . . . .	27
1.2.	Diagrama de Gantt: Desde el 1 de abril de 2024 hasta la primera semana de noviembre de 2024 . . . . .	27
1.3.	Presupuesto de la infraestructura de hardware . . . . .	29
1.4.	Presupuesto de la infraestructura de software . . . . .	29
1.5.	Presupuesto de los Recursos Humanos . . . . .	29
1.6.	Presupuesto total del proyecto . . . . .	29
2.1.	Rutina de boxeo semanal de un gimnasio amateur. . . . .	38
2.2.	FC absoluta media, máxima y mínima por tipo de asalto según nivel competitivo, edad y género [1]. . . . .	40
2.3.	Valores de referencia de FC media por tipo de asalto . . . . .	41
2.4.	Categorías de peso amateur según el reglamento WBC. . . . .	42
2.5.	Niveles Garmin de VO <sub>2</sub> Máximo según sexo y edad. . . . .	44
2.6.	Categorización de Garmin del Body Battery según su valor . . . . .	44
2.7.	Categorización de Garmin del nivel de estrés según su valor . . . . .	45
3.1.	Materiales y características de los componentes físicos del Garmin Venu 2 . . . . .	51
3.2.	Autonomía del Garmin Venu 2 según el modo de uso . . . . .	52
3.3.	Características físicas del Garmin HRM-Pro Plus. . . . .	55
3.4.	Niveles de calificación y su puntuación. . . . .	60
3.5.	Parámetros para la evaluación de la recuperación por tipo de día . . . . .	62
3.6.	Criterio de calificación de los parámetros de evaluación de la recuperación . . . . .	63
3.7.	Tabla de pesos de cada parámetro por tipo de día. . . . .	64
3.8.	Criterio de calificación global de la recuperación . . . . .	64
3.9.	Parámetros para la evaluación del rendimiento deportivo . . . . .	65
3.10.	Criterio de calificación de los parámetros de evaluación del rendimiento . . . . .	66
3.11.	Parámetros para la evaluación del rendimiento deportivo . . . . .	66
3.12.	Criterio de calificación global del rendimiento en función del VO <sub>2</sub> máximo . . . . .	67
3.13.	Aspectos sobre los que se registran sensaciones y observaciones de los usuarios . . . . .	67
3.14.	Orientación de la encuesta SUS a la tecnología wearable . . . . .	68
3.15.	Calificación SUS según la puntuación SUS. . . . .	69
3.16.	Nivel de aceptabilidad según puntuación SUS . . . . .	69
4.1.	Caption . . . . .	72
4.2.	Plan de monitorización de la Fase de Preparación . . . . .	73
4.3.	Plan de monitorización de la Fase Pre-entrenamiento . . . . .	73
4.4.	Plan de monitorización de la Fase de Entrenamiento . . . . .	74
4.5.	Plan de monitorización de la Fase Post-entrenamiento . . . . .	75
4.6.	Plan de monitorización de los días de descanso . . . . .	75
4.7.	Características físicas del usuario 1 . . . . .	78
4.8.	Características físicas del usuario 2 . . . . .	78

## ÍNDICE DE CUADROS

---

5.1. FC y HRV basal por usuario . . . . .	108
5.2. Comparación del VO <sub>2</sub> máximo inicial de los usuarios con el de referencia . . . . .	108
5.3. FC media, máxima y mínima registrada en los entrenamientos, usuario 1 . . . . .	110
5.4. FC media, máxima y mínima registrada en los entrenamientos, usuario 2 . . . . .	110
5.5. HRR y zonas de FC óptima por usuario . . . . .	111
5.6. Comparativa de FC media experimental con zonas de FC óptima . . . . .	111
5.7. Comparativa teórica/experimental de valores medios de FC por tipo de intervalo, usuario 1 . . . . .	112
5.8. Comparativa teórica/experimental de valores medios de FC por tipo de intervalo, usuario 2 . . . . .	112
5.9. Evolución temporal del HRV durante la semana de entrenamiento, usuario 1 . . . . .	115
5.10. Evolución temporal del HRV durante la semana de entrenamiento, usuario 2 . . . . .	115
5.11. Body battery a lo largo de la semana, usuario 2 . . . . .	117
5.12. Frecuencia respiratoria media pre, post y durante el entrenamiento, usuario 1 . . . . .	118
5.13. Frecuencia respiratoria media pre, post y durante el entrenamiento, usuario 2 . . . . .	119
5.14. Nivel de estrés medio por tipo de día a lo largo de la semana, usuario 2 . . . . .	119
5.15. VO <sub>2</sub> máximo inicial y obtenido tras la semana de entrenamientos . . . . .	122
5.16. Sensaciones del usuario 1 durante los entrenamientos . . . . .	126
5.17. Sensaciones del usuario 2 durante los entrenamientos . . . . .	127
5.18. Autonomía del Garmin Venu 2 con el modo de medición por segundo . . . . .	127
5.19. Encuesta SUS de la red personal de dispositivos wearables . . . . .	128
6.1. Tabla de nivel de consecución de los objetivos propuestos . . . . .	130
D.1. Elementos software utilizados en el desarrollo del Proyecto . . . . .	176

# 1. Introducción

## 1.1. Motivación

En un mundo digital en perpetuo crecimiento, es lógico que los humanos busquemos aplicar los avances tecnológicos a nuestras vidas cotidianas para hacerlas más cómodas y eficientes. Redes de automóviles, hogares inteligentes e incluso entornos laborales completos, conectados a la red e intercambiando información en tiempo real, son algunas de las aplicaciones del tan notorio Internet de las Cosas o IoT al día a día de las personas.

De manera análoga a la industria telefónica con la transición de teléfonos fijos a móviles, el factor de movilidad en la tecnología ha hecho que la popularidad y el uso del IoT se multiplique. La portabilidad de los dispositivos inteligentes se traduce en la digitalización de los aspectos más cotidianos de la vida de los usuarios, permitiendo el seguimiento de ámbitos como el ocio, el deporte o la salud.

De esta idea de hacer portable el Internet de las Cosas, nace la tecnología wearable, objeto de estudio de este trabajo debido no solo a su popularidad, sino a la amplia variedad de posibilidades que ofrece. Auriculares inalámbricos que adaptan el nivel de aislamiento acústico automáticamente, camisetas que incorporan en su tejido sensores térmicos, zapatillas que analizan nuestras pisadas, pulsómetros para monitorizar nuestra frecuencia cardíaca o pulseras y relojes inteligentes, todos son ejemplos de dispositivos wearables cuyo funcionamiento seguramente conocemos de primera mano o por un conocido. Y es que la industria del wearable, de la mano de los otros sectores IoT, también está viviendo un gran auge de popularidad.

Este auge se refleja en el crecimiento de su mercantilización. Estudios estadísticos muestran un incremento del 600 % en el número de ventas de smartwatches en la última década, esperándose 258 millones de unidades vendidas en 2025, respecto a los 37 millones registrados en 2016 [2]. La Corporación Internacional de Datos (IDC), por otro lado, proyecta que el mercado de wearables mantendrá un crecimiento de un solo dígito durante varios años, alcanzando los 629.4 millones de unidades en 2027, con una tasa de crecimiento anual compuesta (CAGR) del 5.0 % [3].

Este proyecto también se nutre del auge que están experimentando los deportes de contacto en nuestro país, impulsado por los recientes éxitos de atletas nacionales, como el campeonato mundial de peso pluma de la UFC logrado por Ilia Topuria o la gran campaña de la selección española de boxeo en los Juegos Olímpicos de París. Estos acontecimientos, sumados a la organización de eventos transmitidos por streaming protagonizados por creadores de contenido, ha contribuido significativamente al aumento de afiliados en gimnasios locales dedicados a estas disciplinas.

Por otro lado, numerosos estudios reflejan el impacto positivo de la práctica de deportes de contacto y artes marciales en niños que sufrieron acoso o bullying escolar, reflejado en una mejora del léxico biopsicosocial, el comportamiento prosocial, la ética y la estética de los niños y adolescentes que las practican [4].

Esta creciente demanda hace interesante el desarrollo de aplicaciones y sistemas de análisis de datos que faciliten la experiencia a los usuarios, monitorizando su rendimiento y recuperación ante una actividad tan exigente física y mentalmente.

### 1.2. Objetivos

Esta búsqueda de la mejora de la calidad de vida de las personas se traslada al mundo del deporte y la salud. La demanda de los deportistas de una experiencia personalizada, que respalde con información y datos el progreso de su rendimiento deportivo, cómoda y asequible, que sea permisiva tanto física como económicamente, y sobre todo sencilla e intuitiva, que no suponga una pérdida de tiempo al usuario, ha fomentado el desarrollo de dispositivos wearables que cumplan todos estos requisitos desde la cotidianidad de llevar una prenda o accesorio más.

En este contexto, el objetivo principal del proyecto se centra en la monitorización del rendimiento deportivo y la recuperación de un usuario en deportes de contacto mediante el despliegue de una red personal de dispositivos wearables y la explotación de los datos capturados mediante la realización de un análisis de dichos datos.

Para cumplir el objetivo principal se tienen que realizar los siguientes objetivos específicos:

- Explorar los distintos tipos de dispositivos wearables que pueden utilizarse para montar una red personal de dispositivos que cumpla con los requisitos de los deportes de contacto.
- Determinar las métricas a medir con la red de dispositivos wearables para tener información valiosa de la recuperación y el rendimiento en el entrenamiento de un deporte de contacto.
- Poner a punto la red corporal de dispositivos así como la infraestructura utilizada para la captación de datos durante el entrenamiento de deportes de contacto.
- Evaluar que los dispositivos utilizados en la red personal sean pocos invasivos en la práctica del deporte de contacto y tengan la capacidad de monitorizar las métricas más adecuadas.
- Analizar y establecer estrategias para el análisis de los datos con objeto de proporcionar información valiosa sobre el rendimiento y la recuperación de los usuarios, que pudiera ser usada por especialistas del deporte o la salud para tomar las decisiones pertinentes.
- Aplicar el sistema desarrollado así como la analítica realizada para el caso de estudio de deportes de contacto.
- Realizar un piloto con un conjunto amplio de sujetos de estudio para validar el sistema y los algoritmos desarrollados.
- Utilizar buenas prácticas de desarrollo de software en el proceso de desarrollo y puesta a punto de la infraestructura empleada para la monitorización de la actividad deportiva.

### 1.3. Planificación

Esta sección describe las fases en las que se divide la planificación del Proyecto, así como su disposición temporal, mostrada en un diagrama de Gantt.

### **1.3.1. Fase 1: Planteamiento de los objetivos**

En la fase inicial del Proyecto se llevaron a cabo reuniones con el tutor para especificar los objetivos y pautas que el objetivo debería seguir originalmente. La poca investigación del campo de los deportes de contacto, por su reciente auge, hizo que estos objetivos se fueran moldeando durante la fase de documentación.

Esta fase se tuvo que repetir avanzado el desarrollo, para volver a concretar estos objetivos tras la no consecución de uno de los originales (Fase 3.2).

### **1.3.2. Fase 2: Documentación teórica del Proyecto**

La segunda fase del proyecto se dedicó al estudio de la base teórica del mismo, con el objetivo de fundamentar los objetivos y seleccionar los materiales del Proyecto. Se dividió en tres sub-fases:

#### **Fase 2.1: Estudio de la tecnología wearable**

El de búsqueda comenzó con el estudio de los principios fundamentales de los dispositivos wearables y las redes personales.

Este proceso se retomó una vez se seleccionaron los dispositivos específicos a utilizar en el Proyecto tras la Fase 1.3, pasando a analizar la infraestructura de su compañía desarrolladora.

#### **Fase 2.2: Revisión de la literatura de la ciencia de los deportes de contacto**

La siguiente fase de la documentación tuvo como objetivo la caracterización temporal y en parámetros fisiológicos (medibles por wearables) de los deportes de contacto, analizando los estudios de la literatura del campo.

#### **Fase 2.3: Estudio de mercado de wearables**

La última fase de la documentación consistió en el estudio del mercado de los dispositivos wearables para la elección de red personal adecuada a los requerimientos de los deportes de contacto.

La duración de esta fase fue de 10 semanas, divididas en 4 meses por un parón en el periodo de exámenes.

### **1.3.3. Fase 3: Desarrollo experimental y técnico**

La siguiente fase del Proyecto consiste en el desarrollo del proceso de monitorización y de su implementación en código. Se divide en dos fases principales:

#### **Fase 3.1: Desarrollo del plan de monitorización y el algoritmo de calificación**

En esta fase se ideó el plan de monitorización con dispositivos wearables a seguir por los usuarios, así como el sistema de calificación de la recuperación y el rendimiento.

Se tuvo que replantear por el objetivo fallido en la Fase 3.2.

### **Fase 3.2: Implementación del plan de monitorización**

Esta fase corresponde a la implementación en código del planteamiento propuesto para la monitorización, incluyendo la familiarización con los entornos de programación empleados.

Esta fase se divide en dos etapas, debido a que se abandonó primer foco de desarrollo por una serie de inconvenientes documentados en el Capítulo 4 de esta memoria.

Tras el replanteamiento de los objetivos, se continuó con el desarrollo del programa de depurado, análisis y visualización de los datos capturados.

#### **1.3.4. Fase 4: Captura de los datos de estudio**

La cuarta fase se dedicó a la captura de los datos de monitorización de la recuperación y el rendimiento de los dos usuarios del estudio, siguiendo el plan desarrollado.

#### **1.3.5. Fase 5: Análisis de los datos capturados**

La fase 5 del trabajo es la referente al procesado de los datos capturados. Se divide en dos sub-fases:

##### **Fase 5.1: Depurado y análisis de los datos**

La primera fase del procesado corresponde a la obtención del DataSet de análisis tras el depurado y el ordenamiento manual de los datos, para la obtención de gráficas y resultados en el proceso de análisis.

##### **Fase 5.2: Calificación del rendimiento y la recuperación con el algoritmo propuesto**

La segunda fase consiste en el análisis de los resultados al aplicar el algoritmo de calificación planteado a los datos tras su análisis.

#### **1.3.6. Fase 6: Redacción del informe**

La última fase del informe es la relativa a la confección de esta memoria, asegurando que el proceso de desarrollo y análisis de los resultados están bien documentados.

	Octubre 2023				Noviembre 2023				Diciembre 2023				Febrero 2024				Marzo 2024				
Fases	02	09	16	23	06	13	20	27	04	11	18	25	05	12	19	26	04	11	18	25	
Fase 1	■												■								
Fase 2		■								■				■							
Fase 2.1		■											■								
Fase 2.2			■																		
Fase 2.3				■						■											
Fase 3														■							
Fase 3.1														■							
Fase 3.2														■							
Fase 4																					
Fase 5																					
Fase 5.1																					
Fase 5.2																					
Fase 6																					

Cuadro 1.1: Diagrama de Gantt: Octubre 2023 al 25 de marzo de 2024

	Abril 2024				Mayo 2024				Agosto 2024				Septiembre 2024				Octubre 2024				Noviembre 2024	
Fases	01	08	15	22	05	12	19	26	02	09	16	23	01	08	15	22	29	07	14	21	28	04
Fase 1	■												■									
Fase 2																						
Fase 2.1																						
Fase 2.2																						
Fase 2.3																						
Fase 3																						
Fase 3.1																						
Fase 3.2																						
Fase 4														■				■				
Fase 5														■				■				
Fase 5.1														■				■				
Fase 5.2																						
Fase 6																						

Cuadro 1.2: Diagrama de Gantt: Desde el 1 de abril de 2024 hasta la primera semana de noviembre de 2024

## 1.4. Componentes utilizados

Esta sección expone el equipo, compuesto por hardware, software y recursos humanos, con el que se ha trabajado para el desarrollo del proyecto.

### 1.4.1. Hardware

- Ordenador Portátil ASUS Zenbook UX431F con procesador Intel Core i7-10510U.
- Reloj inteligente Garmin Venu 2, con funda de protección y cable de conexión USB.
- Sensor de frecuencia cardíaca Garmin HRM-Pro Plus.

### 1.4.2. Software

En esta sección se listan los componentes software principales. La lista completa se recoge en el apéndice ??.

- Sistema Operativo Windows 11.
- Distribuidor de Python Anaconda Navigator.
- Overleaf LATEX.

### 1.4.3. Recursos humanos

- D. Juan Antonio Holgado Terriza, coordinador del Master Universitario En Desarrollo de Software de la Escuela Internacional de Postgrado y profesor titular de la Universidad de Granada en el Departamento de Lenguajes y Sistemas Informáticos, como tutor del proyecto.
- Marcos Tendero Carmona, alumno del Grado en Ingeniería de Tecnologías de Telecomunicación con mención en Telemática, como autor del proyecto.

## 1.5. Presupuesto

En esta sección se recoge el presupuesto con el que se cuenta para la realización del proyecto, en el que se incluyen los costes del equipo hardware y software, así como el precio de los avales del personal encargado según las horas de trabajo dedicadas.

### 1.5.1. Presupuesto de la infraestructura de Hardware

Este apartado recoge los costes asociados al equipo hardware con el que se ha trabajado en el proyecto.

### 1.5.2. Presupuesto de la infraestructura de Software

Este apartado hace referencia al coste de todas las plataformas software utilizadas para la elaboración del proyecto.

Recursos Hardware	Subtotal
Ordenador portátil ASUS Zenbook UX431F (Amortización de 360 horas)	300,00 €
Garmin Venu 2	399,99 €
Garmin HRM-Pro Plus	129,99 €
<b>TOTAL</b>	<b>829,98 €</b>

Cuadro 1.3: Presupuesto de la infraestructura de hardware

Recursos Software	Descripción	Subtotal
Sistema operativo Windows 10 pro	Pre-instalada	0,00 €
Distribuidor Anaconda Navigator	Gratuito	0,00 €
Overleaf	Gratuito	0,00 €
<b>TOTAL</b>	-	<b>0,00 €</b>

Cuadro 1.4: Presupuesto de la infraestructura de software

### 1.5.3. Presupuesto recursos humanos

En esta sección se calcula el coste total de los recursos humanos que intervienen en el proyecto, según el número de horas empleadas, considerando el costo por hora de un alumno del Grado en Ingeniería de Tecnologías de Telecomunicación y de un profesor titulado de la Universidad de Granada.

Recursos Humanos	Precio	Duración	Subtotal
Alumno de Ingeniería de Telecomunicaciones	20€/hora	490 horas	9800,00€
Tutor (Profesor Titulado UGR)	40€/hora	36 horas	1440,00€
<b>TOTAL</b>	-	-	<b>11240,00 €</b>

Cuadro 1.5: Presupuesto de los Recursos Humanos

### 1.5.4. Presupuesto Total

En este último apartado se recoge el desglose del presupuesto total del proyecto.

Recursos Humanos	Precio
Hardware	829,98 €
Software	0,00€
Recursos Humanos	11240,00€
<b>TOTAL</b>	<b>12069,98 €</b>

Cuadro 1.6: Presupuesto total del proyecto

## 1.6. Estructura del Proyecto

El informe del Proyecto está organizado en una estructura de seis capítulos principales, respaldados por una serie de índices y una sección para las referencias bibliográficas utilizadas. En esta sección se introducen con brevedad estos capítulos.

- **Capítulo 1: Introducción.** El presente capítulo recoge las secciones que introducen al Proyecto, referentes a la motivación que lo sustenta, los objetivos propuestos, la planificación seguida durante su desarrollo y la estructura de este informe. Además, se listan los componentes utilizados para su realización y su coste asociado.

- **Capítulo 2: Estado del Arte.** Este capítulo introduce la tecnología wearable y explora de manera detallada, a modo de revisión, la literatura de las ciencias de la salud y el deporte focalizada en los deportes de contacto, con objetivo de exponer al lector cómo estos dispositivos pueden mejorar el rendimiento y la recuperación del deportista.
- **Capítulo 3: Materiales y Métodos.** El tercer capítulo del documento describe las características del equipo, tanto hardware como software, con el que se trabaja a lo largo del desarrollo del Proyecto.
- **Capítulo 4: Desarrollo Experimental y Técnico.** En este capítulo se detallan los procedimientos seguidos para el desarrollo del Trabajo. Se incluye tanto el proceso experimental seguido por los usuarios para la captación de datos como el desarrollo completo del análisis de los mismos.
- **Capítulo 5: Análisis de Resultados.** Este capítulo esta reservado al análisis de los datos obtenidos a partir del desarrollo técnico y experimental, fundamentado con los estudios científicos de la literatura de la ciencia del deporte y la salud, revisados en el segundo capítulo del documento.
- **Capítulo 6: Conclusiones y Líneas Futuras.** Finalmente, se incluye un capítulo para las conclusiones alcanzadas tras el desarrollo del Proyecto y el análisis de los resultados obtenidos.

Se incluye un apartado con vista a futuro, comentando posibles mejoras y aspectos a tener en cuenta para posibles investigaciones en el mismo campo.

## 2. Estado del Arte

### 2.1. Dispositivo Wearable

Un dispositivo wearable, o ponible, es un tipo de dispositivo que está diseñado, como su propio nombre indica, para estar en contacto con el cuerpo, con el objetivo de obtener información relativa a la persona en cuanto a su actividad o su salud. Analizando su arquitectura, podemos considerar que los wearables son la aplicación del Internet de las Cosas a la vestimenta.

Por otro lado, una red personal o red de área personal (PAN), es un sistema de monitorización wearable compuesto por varios dispositivos, controlados por un dispositivo principal, de manera que su uso conjunto permita complementar las funcionalidades de los dispositivos por separado.

#### 2.1.1. Características del Dispositivo Wearable

A continuación, describimos las características fundamentales que definen a este tipo de tecnología [5]:

- **Portabilidad y durabilidad:** Su diseño se basa en la búsqueda de una mejora de salud sin salir de la vida cotidiana, de manera que llevarlos puestos no interfiera en el resto de actividades. Están diseñados para ser llevados puestos durante el día, total o parcialmente, por lo que tienen que ser lo más cómodos, livianos, robustos y flexibles posibles, además de fáciles de lavar.
- **Autonomía:** Estos dispositivos están diseñados para operar con la mínima interacción del usuario, haciendo un seguimiento general autónomo de su salud y rendimiento, aunque también cuenten con la capacidad de medir parámetros específicos mediante aplicaciones concretas activadas de manera manual. Por ello, incorporan baterías con la capacidad suficiente para poder operar durante días de manera continua, lo que supone un reto debido al tamaño reducido de los dispositivos.
- **Capacidad de Monitorización:** Los dispositivos wearables están equipados con numerosos sensores, diseñados para la monitorización autónoma de parámetros fisiológicos o de actividad en tiempo real, para su análisis y control, con fines médicos o de rendimiento.
- **Conectividad:** Estos dispositivos cuentan con sistemas de comunicación inalámbrica que permiten la visualización de los datos en interfaces externas (en dispositivos móviles o aplicaciones web) o para la comunicación M2M, creando redes personales de varios dispositivos. Para ello, emplean protocolos de conectividad inalámbrica, estandarizados como BLE, Wi-Fi o NFC, o propios como ANT.

### 2.1.2. Arquitectura de un sistema de monitorización con dispositivos wearables

La arquitectura de los dispositivos portátiles se puede conceptualizar en capas separadas, que se centran en diferentes aspectos de la gestión de datos, desde la captura hasta la visualización del dato. Aquí describimos cómo organizar estas capas [6]:

- **Capa de captación de datos:** En esta capa se encuentran los distintos sensores con los que se extraen los datos del entorno: sensores de movimiento, sensores biométricos o sensores ambientales son todos ejemplos de la tecnología que incorporan los dispositivos wearables para la adquisición de datos. La literatura sobre dispositivos wearables e IoT incluye una capa intermedia entre esta y la siguiente (almacenamiento), correspondiente a los mecanismos de pasarela o gateway de los datos desde su recogida a su almacenamiento, mediante el uso de protocolos inalámbricos de transferencia de datos, como BLE o ANT (que comentaremos más adelante). En este proyecto se incluye como subcapa, debido a que se considera este procedimiento implícito en la recuperación de datos.
- **Capa de almacenamiento:** Esta capa corresponde al almacenamiento de los datos capturados por los sensores, además de la gestión de acceso para su posterior importación y análisis.

Podemos diferenciar entre almacenamiento local, en la memoria de los dispositivos wearables (por las características que hemos explicado anteriormente, tienen una memoria limitada, por lo que se suelen guardar pocos datos para ser mostrados por pantalla) o en dispositivos móviles, o en la nube, que es la opción más popular para las grandes compañías.

- **Capa de análisis:** Esta capa recoge las técnicas de procesamiento del bruto de datos para extraer la información de interés, como la evolución del estado de un paciente, patrones de comportamiento o incluso predicciones, mediante algoritmos o modelos de procesado de datos. Se puede diferenciar entre dos tipos de análisis, según la inmediatez del procedimiento:
  - Análisis en tiempo real: analiza los datos para devolver resultados de la forma más rápida y eficaz posible, como la frecuencia cardíaca en tiempo real durante una actividad deportiva.
  - Análisis profundo: utiliza algoritmos más sofisticados para realizar análisis corporales y devolver datos más complejos, como identificación de patrones o tendencias.
- **Capa de visualización o aplicación:** Esta capa se encarga de presentar la información ya procesada y analizada a los usuarios por la interfaz, según la funcionalidad de la aplicación diseñada. Recoge mecanismos como los sistemas de envío de notificaciones y de interacción con el usuario.

Estas aplicaciones no siempre son gestionadas por el propio usuario del dispositivo de monitorización, sino que entidades externas, como médicos o entrenadores personales, podrán encargarse de analizar los resultados y tomar decisiones sobre recomendaciones para nuestra rutina o tratamiento.

Finalmente, representamos esta arquitectura de capas de manera gráfica en la siguiente figura:

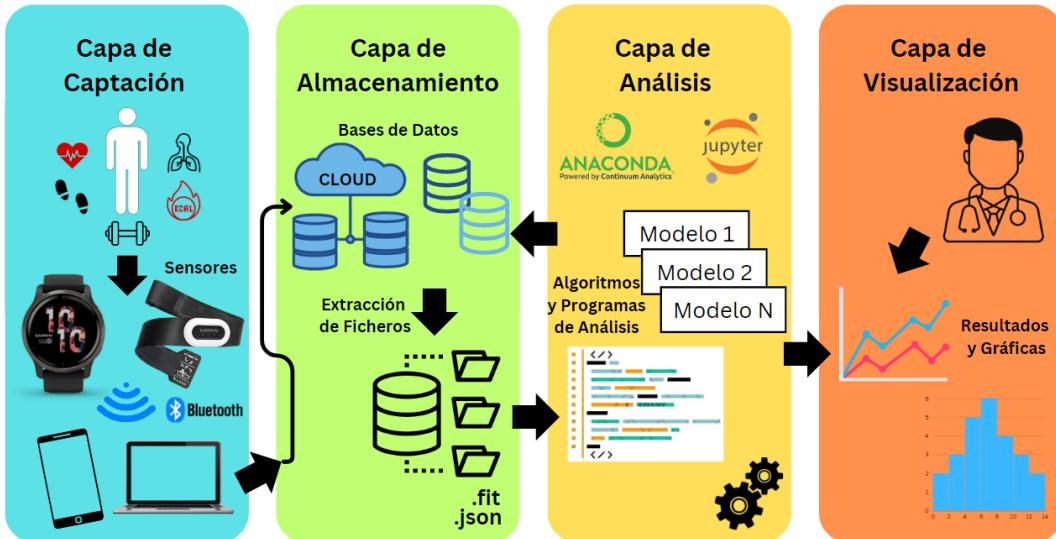


Figura 2.1: Arquitectura de un sistema de monitorización con wearables. Elaboración propia.

### 2.1.3. Tipos de wearables

En esta sección se revisarán, a modo de búsqueda, los principales tipos de dispositivos wearables ofertados en el mercado, para poder posteriormente seleccionar el más (o los más, formando una red personal) adecuado para nuestro caso de estudio. Como el proyecto se centra en el deportista amateur, incluiremos su rango de precio de mercado, buscando que sean asequible de acuerdo a las prestaciones que ofrece.

Los dispositivos wearables más populares para la monitorización deportiva son los siguientes:

- **Smartwatches.** Son el dispositivo vestible por excelencia. Los relojes inteligentes añaden a la funcionalidad del reloj común la posibilidad de medir de manera continua parámetros fisiológicos, como las frecuencias cardíaca y respiratoria, saturación de oxígeno en sangre, edad fisiológica, etc., y la monitorización de la actividad física con aplicaciones deportivas, todo de manera personalizada mediante la configuración de un perfil de usuario. Pueden ser autónomos o conectados a un dispositivo externo, como el móvil, para su gestión.

En el mercado, su precio oscila entre los 200 y los 1000 euros.

- **Pulseras inteligentes.** Las pulseras de actividad son una versión simplificada (y por ende más económica) de los relojes, no solo en su reducida (o inexistente) pantalla, sino en las funcionalidades ofrecidas. Esta limitación hace que la mayoría de modelos no cumplan los requisitos de este estudio, principalmente porque están diseñados para un público que se ejerce de manera casual.

Su precio oscila entre los 25 y los 200 euros.

- **Anillos inteligentes.** Quizás el más sorprendente de la lista, debido a lo desapercibido que pasa o, más bien, por lo poco invasivo que es su uso. Permite, mediante el control con una aplicación externa, medir una gran cantidad de parámetros fisiológicos por sensores ópticos en el dedo. Su popularidad creció en el deporte de élite al ser empleados por la NBA para la detección del covid-19 en sus jugadores durante la pandemia de 2020.

Su precio ronda los 200-300 euros.

- **Gafas inteligentes.** Tanto de realidad virtual, como de realidad aumentada o asistida. Estas últimas son las más interesantes, ofreciendo, pese a su ligero peso, no solo corrección

de visión, sino la posibilidad de reproducir contenido multimedia y de aplicaciones móviles empleando microproyectores led, micrófono y altavoces.

La encontramos desde los 200 hasta los 3500 euros.

- **Prendas inteligentes.** Desde camisetas que miden los componentes del sudor y la temperatura corporal, hasta calzado, como zapatillas, calcetines o suelas, que monitorizan los puntos de apoyo de las pisadas o la distancia de las zancadas en una carrera, este tipo de wearable incluyen sensores incrustados en el tejido para la monitorización de diversos parámetros de una manera cómoda e inadvertida.
- **Sensores complementarios.** Referentes a todos los sensores que se conectan de manera inalámbrica a los dispositivos wearables para complementar su funcionalidad, añadiendo un parámetro que antes no medían o mejorando la precisión respecto a la que ofrecían por defecto. Es el caso (entre otros) de sistemas de posicionamiento GPS para bicicletas, acelerómetros para velocistas enganchables en zapatillas o, como veremos en este proyecto, sensores de frecuencia cardíaca equipados en bandas pectorales.

El precio de estos sensores es muy variado. En el caso de las bandas pectorales, rondan los 70-130 euros.

### 2.1.4. Análisis del mercado de dispositivos wearables. Diseño de una red personal

En esta sección se describe el procedimiento de análisis del mercado del sector para la realización del proyecto.

Una vez conocidas las características fundamentales de los dispositivos wearables y las prestaciones que nos brinda cada tipo de dispositivo para la monitorización de los usuarios, debemos plantear cuál es la red personal más adecuada para cubrir los requerimientos que impone la práctica de deportes de contacto.

Estos requerimientos son los siguientes:

- **Requerimientos de parámetros.** El primer objetivo de la red personal desplegada es que pueda medir todos los parámetros que se exponen sección 2.2.3. Además, procuraremos que la medición de dichos parámetros sea la más precisa posible, criterio clave en la selección de los dispositivos.
  - **Requerimientos físicos.** Referidos a las limitaciones físicas implícitas en el propio deporte. Algunos de los sensores de los dispositivos wearables son sensibles al movimiento, por lo que, por ejemplo, la utilización de un smartwatch, colocado en la muñeca, se ve capada en una actividad en la que se muevan mucho los brazos, como es el caso de la mayoría de deportes de contacto.
- Por otro lado, buscaremos un dispositivo lo suficientemente resistente para aguantar posibles golpes y roces, así como que sean cómodos a la hora de llevarlos puestos durante la actividad.
- **Asequibilidad económica.** Debido a que el estudio se centra en usuarios amateur, valoraremos que el coste del sistema sea lo más permisivo posible en términos de coste.

El análisis comenzó analizando qué tipo de dispositivos cumplían con los requerimientos de parámetros. Se concluyó que smartwatches, pulseras inteligentes y anillos inteligentes como dispositivos principales, en combinación con sensores independientes de frecuencia cardíaca, pasaban estos filtros.

Valorando los requisitos físicos del dispositivo, se descartó el uso de anillos inteligentes, ya que el impacto de los dedos en la práctica de deportes de contactos impide su uso durante el entrenamiento.

Haciendo una cuantización de funcionalidades por coste económico, los candidatos finales fueron la pulsera inteligente Fitbit Charge 3 y el smartwatch Garmin Venu 2 como dispositivo principal y las bandas Polar H10 y Garmin HRM-Pro Plus como sensor de FC externo.

Finalmente, la infraestructura de gestión y almacenamiento de datos, así como la garantía que brinda ser la mayor empresa de wearables del mercado, ha inclinado la balanza a los dispositivos Garmin.

Así, se propone una **red personal compuesta por el smartwatch Venu 2 como dispositivo principal y la banda con sensor de frecuencia cardíaca HRM-Pro Plus**: El reloj inteligente de Garmin nos permitirá hacer el seguimiento de todos los parámetros externos al entrenamiento y gestionará la monitorización del mismo. Por otro lado, la banda Garmin HRM-Pro Plus nos permitirá mediciones de FC de alta precisión durante el entrenamiento, ya que está diseñada el uso deportivo.

## 2.2. Caracterización del entrenamiento de deportes de contacto

Con el fin de elegir un wearable o red personal adecuado los requisitos derivados de la práctica de un deporte de contacto, debemos conocer, no solo las propias características del propio deporte y los parámetros monitorizables durante su realización, sino qué otros valores de interés podemos medir fuera del periodo de actividad, permitiéndonos alcanzar el mayor nivel de rendimiento posible y una recuperación óptima.

### 2.2.1. Estructura temporal y caracterización física

El reglamento del Consejo Mundial de Boxeo (CMB) dicta en su normativa [7] que las competiciones amateur se regirán con una temporización de 3 rounds o asaltos de 3 minutos con descansos intra-round de 1 minuto para púgiles masculinos, y de 3 rounds de 2 minutos con descansos de 1 minuto para boxeadoras. Por otro lado, la FEK determina que los combates amateur seguirán un formato de 3 asaltos de 2 minutos con 1 minuto de descanso entre asaltos [8]. Esta estructura temporal determina, por otro lado, la manera de preparar a los atletas, y se traslada a los entrenamientos en los gimnasios, rigiendo su funcionamiento. Estos gimnasios juntan tanto a competidores como a practicantes regulares, por lo que cualquier aficionado, objetivo parcial de este estudio, seguirá este plan de entrenamiento. Es el caso del gimnasio granadino de boxeo M&S, regentado por el campeón europeo Francisco Martos, en el que encontramos practicantes de todos los niveles y donde un reloj programado en intervalos de 3 minutos para la actividad y 1 minuto para los de recuperación dicta el funcionamiento de las clases.

En cuanto a la caracterización física de la actividad, tanto en competición como entrenamientos, la alta intensidad física de los asaltos, en los que se lanzan y reciben golpes en continuo movimiento, categoriza a las artes marciales como deportes interválicos de potencia aeróbica [9]. Por tanto, la capacidad de recuperación, tanto durante el propio asalto (pasiva, por cese de actividad) como en el minuto de descanso (activa) será un factor totalmente condicional en nuestro rendimiento [10]. Este cansancio no solo afecta al desempeño deportivo de manera directa, sino que mandará un mensaje de debilidad a nuestro rival, que podrá capitalizar la situación para puntuar, introduciendo el factor mental del deporte.

Tras este análisis, podemos definir la naturaleza temporal y física del entrenamiento pugilístico como **un entrenamiento interválico de alta intensidad en bloques de acciones**

acíclicas con intervalos de descanso [10]. Esta descripción encaja por sus propiedades con el tan conocido entrenamiento interválico de alta intensidad, comúnmente llamado HIIT por sus siglas en inglés (High Intensity Interval Training), que está altamente desarrollado en relojes inteligentes y que usualmente se instala por defecto debido a su demanda para todo tipo de entrenamientos.

Esta caracterización física y temporal es crucial, ya que nos da a priori información de valor sobre **cuándo podremos medir según qué parámetros** durante el periodo en el que transcurre el entrenamiento. Las sensibilidad de ciertos sensores y algoritmos al movimiento requieren que el usuario se halle en reposo para la medición de ciertos parámetros (como es el caso del estrés o la frecuencia respiratoria). Como se ha comentado, los intervalos de actividad albergan una alta actividad física, por lo que contaremos únicamente con los periodos de recuperación para la obtención de cierta información.

### 2.2.2. Rutina de boxeo de un gimnasio amateur

En este apartado se desarrolla el plan de entrenamiento que seguirán los practicantes objetivo de este estudio, impartido en el gimnasio de boxeo M&S Boxeo en granada, regido por Francisco Martos, boxeador profesional de mayor edad en conseguir un título del Consejo Mundial de Boxeo. En concreto, analizaremos la rutina de un usuario de nivel principiante (o no orientado a competición), ya que constituyen el mayor porcentaje de alumnos de un gimnasio regular.

Siguiendo lo comentado en el apartado anterior sobre la normativa amateur del WBC, el entrenamiento se basa temporalmente en rounds de actividad de 3 minutos con periodos de recuperación intra-round de 1 minuto. En este caso, la sesión de entrenamiento durará 1 hora y media.

La rutina se divide principalmente en cinco partes o secciones, que se reparten en 4 sesiones y un día descanso, de lunes a viernes. Estas son: calentamiento, tecnicificación individual, tecnicificación coordinada o en parejas, preparación física específica y sparring.

Para cada una de las secciones, especificaremos el número de **Asaltos de Actividad (AA)** y de **Asaltos de Recuperación (AR)** que se les dedicará en cada sesión de entrenamiento.

■ **Calentamiento:** El calentamiento consta de 7 rounds de actividad (AA) y 7 de recuperación (AR), sumando un total de 24 minutos de entrenamiento, casi 1/3 de la actividad total. Esto se debe a la exigencia física del resto de entrenamiento dada la explosividad de los ejercicios, que requieren que nuestros músculos y articulaciones estén calientes. Será común a todos los entrenamientos y se compone de los siguientes ejercicios:

- **Comba.** El salto a la comba presenta amplios beneficios para los boxeadores, desde una mejora de nuestra resistencia cardiovascular, de coordinación, equilibrio y agilidad, hasta una mejora muscular y ósea del tren inferior del cuerpo [11]. Cada una de las sesiones se dedican 3 asaltos al salto a la comba, usando también los periodos de descanso como periodos de actividad, lo que se traduce en 12 minutos de comba ininterrumpidos.
- **Movilidad.** Ejercicios centrados en calentar el tren inferior y articulaciones superiores para prepararlas para el entrenamiento, así como mejorar nuestra coordinación. Para trabajar este aspecto, se realiza un asalto (+ 1 AR) de desplazamiento por el ring o de movimiento de cadera esquivando una cuerda.
- **Sombra.** Este tipo de actividad consiste en la práctica de golpes ‘al aire’ o sin objetivo (contra ‘tu sombra’), buscando realizar una buena técnica aumentando progresivamente la velocidad de ejecución, y en este caso ayudándonos a calentar músculos

y articulaciones superiores. El uso de resistencia con mancuernas ligeras ayudará además a aumentar nuestra velocidad, simulando el peso de unos guantes de boxeo. Se dedican un total de 2 AA y 2 AR a esta actividad, alternando el tipo de golpes y de pesos.

- **Tecnificación:** La tecnificación es, después del calentamiento, la parte más importante de esta rutina, ya que es la base deportiva de los atletas. Desde a prevenir lesiones musculares y articulares [12], hasta el aprovechamiento de la biomecánica del cuerpo para aumentar la potencia de los golpes, una buena técnica es lo que diferencia a los mejores boxeadores. Distinguimos entre tecnificación individual o por parejas:

- Tecnificación individual. Para entrenar este campo, haremos distintos trabajos de saco. De manera común a todos los entrenamientos haremos, tras el calentamiento, 5 asaltos (+ 5 AR) en el saco variando el tipo de golpes lanzados, a modo de calentamiento de manos y muñecas.

Además, como último ejercicio de todos los entrenamientos, cargando con toda la fatiga acumulada en el mismo, haremos un asalto de un ejercicio de potencia intervalica en el que, siguiendo el pitido de un reloj, trabajamos en el saco cambiando la frecuencia y la fuerza de los golpes en intervalos de 15 segundos dentro del round de 3 minutos. Este ejercicio busca entrenar la exigencia de intensidad con cansancio, mejorando la capacidad cardiovascular de los alumnos.

- Tecnificación por parejas. Aunque el boxeo es un deporte individual, se practica en oposición a un rival, por lo que entrenar la defensa y el contragolpe es tan necesario como la técnica de golpeo. Por tanto, gran parte del entrenamiento, repetido de manera diaria (menos el día de tecnificación individual y de físico) se dedica a entrenar junto a una pareja.

Se practican ejercicios de defensa, combinaciones de golpes y contragolpes, además de juegos de coordinación, lo que permite mejorar el equilibrio, la estabilidad y los reflejos. Cada entrenamiento cuenta con 5 asaltos (AA + AR) de tecnificación por parejas.

- **Trabajo físico:** Un día a la semana se dedica al acondicionamiento físico específico para deportes de contacto, realizando ejercicios que buscan mejorar la fuerza, explosividad y resistencia de músculos y articulaciones.

Se hace mucho énfasis en los hombros, caderas y piernas, debido a que son, biomecánicamente, la base del poder de nuestros golpes [13]. También buscaremos fortalecer nuestro tronco y cuello, que nos hará más resistentes a golpes [14]].

Los ejercicios de fuerza se combinan con circuitos de coordinación de pies, indispensables para mantener una postura estable en el ring en todo momento.

- **Sparring Ligero:** Por último, se practican ejercicios de intercambio de golpes, al cuerpo o ligeros a la cabeza, llegando a hacer simulaciones de combate o ‘sparring’. Estos ejercicios se realizan de manera puntual para principiantes, aumentando la frecuencia cuanto mayor sea la experiencia de los atletas.

Durante estos asaltos (3 AA + AR), los usuarios se equipan de protecciones (casco y protector bucal), para evitar lesiones en la mayor medida posible. El número de asaltos varía dependiendo del entrenamiento.

- **Pausa para Vendaje:** Aparte de los ejercicios comentados, se realiza una pausa entre el calentamiento y la tecnificación para el equipamiento del vendaje protector y los guantes. Este procedimiento conllevará dos asaltos (2 AA + AR), debido a que el usuario ha de asegurarse de que los wearables quedan bien protegidos por dicho equipamiento.

Una vez explicados los distintos ejercicios de los que se compone una sesión de entrenamiento de boxeo en un gimnasio amateur, se expone la rutina semanal sobre la que basaremos el estudio, para posteriormente planificar la metodología de monitorización con dispositivos wearables que seguirá el usuario.

Esta rutina cuenta con 4 días de entrenamiento y 1 de descanso, más el fin de semana. Los días de entrenamiento implementan los ejercicios comentados anteriormente como se muestra a continuación (Ejercicio (AA + AR)):

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Calentamiento	Comba (3 AA+AR)	Descanso		Comba (3 AA+AR)	Descanso		
	Sombra y Movilidad (4 AA + 4 AR)			Sombra y Movilidad (4 AA + 4 AR)			
	Vendaje (2 AA+AR)			Vendaje (2 AA+AR)			
Tecnificación	Trabajo de Saco (5 AA+AR)	Trabajo de Saco (5 AA+AR)		Trabajo de Saco (5 AA+AR)	Trabajo Físico		Descanso
	Trabajo en Parejas (5 AA+AR)	Trabajo en Parejas (5 AA+AR)		Trabajo en Parejas (5 AA+AR)		(10 AA+AR)	
Sparring /Tecnific.	Trabajo de Saco (3 AA+AR)	Trabajo de Saco (3 AA+AR)		Trabajo de Saco (3 AA+AR)	Trab. Saco / Sparring (3 AA+AR)		
	Ejercicio de Potencia Interválica (1 AA+AR)				Ejercicio de Potencia Interválica (1 AA+AR)		

Cuadro 2.1: Rutina de boxeo semanal de un gimnasio amateur.

### 2.2.3. Parámetros de interés medibles con wearables

La monitorización del rendimiento físico y la recuperación de deportistas de contacto con wearables se respalda de la literatura de la ciencia de esta rama del deporte que, aunque limitada, converge en una lista de parámetros fisiológicos precisamente medibles por este tipo de dispositivos.

Aparte de estas variables, que afectan al rendimiento pugilístico de manera directa, el estudio se centrará en aspectos externos a la propia actividad de los que dependerá la recuperación del atleta, que de igual manera tendrán un fuerte impacto en la eficiencia deportiva de los atletas.

- **VO<sub>2</sub> máximo.** Parámetro que mide la cantidad máxima de oxígeno que el organismo puede absorber, transportar y consumir en un periodo determinado, en nuestro caso, durante la práctica de actividades agotadoras o ‘máximas’. Nos permite valorar la capacidad del organismo de combatir la fatiga en una actividad de cierta duración e intensidad física. Define la potencia (o metabolismo) aeróbica máxima, y guarda una relación directa con el rendimiento deportivo de los boxeadores, debido a la demanda metabólica del combate,

suponiendo la capacidad de recuperación entre asalto de vital importancia [6] [15]. Por ello, ha sido el parámetro foco de la literatura sobre el rendimiento boxístico.

Se expresa de forma absoluta (ml/min) y relativa (ml/min/kg). En este estudio se trabajará con la segunda, usada por la mayoría de artículos de investigación de este campo.

Los estudios que ofrece la literatura de deportes de contacto acerca de la importancia del VO<sub>2</sub> máximo en el rendimiento de los deportistas nos brinda información de interés que nos guiará al desarrollo de este estudio. En primer lugar, hay evidencia de que el ejercicio de alta intensidad practicado en el boxeo aumenta los valores de VO<sub>2</sub> Máximo de perfiles sanos no atléticos [16], lo que conlleva que la medida inicial que registremos antes del periodo de recolección de datos no será el máximo real, sino que este valor podrá aumentar.

Por otro lado, otro estudio refleja que la cantidad requerida de consumo máximo de oxígeno relativo para los ejercicios típicos de los que se compone una sesión de entrenamiento de boxeo, comentados en el apartado anterior, es de  $62,2 \pm 4,1$  ml/min/kg [17].

Los principales estudios que han analizado el VO<sub>2</sub> máximo de grupos de boxeadores masculinos de nivel amateur ( $21.4 \pm 3.0$  años) obtienen los siguientes resultados (ml/min/kg): 58.03 [18], 56.6 [19],  $54.5 \pm 4.5$  [20],  $61.7 \pm 9.0$  [21],  $63.8 \pm 4.8$  [22],  $59.5 \pm 4.7$  [23].

Establecemos por tanto, a partir del rango que nos establece este conjunto de valores, que el valor ideal de VO<sub>2</sub> máximo para un boxeador amateur está entre  $54.5 \pm 4.5$  y  $63.8 \pm 4.7$  ml/min/kg. Usaremos este valor para hacer una adaptación a nuestro caso de estudio de los valores de VO<sub>2</sub> máximo tabulados por las empresas de wearables seegún el nivel atlético de sus usuarios.

- **Frecuencia cardíaca (FC) y su variabilidad (HRV).** La monitorización de la FC y su variabilidad son muy interesantes para nuestro estudio. Por un lado, el ritmo de nuestros latidos, o frecuencia cardíaca, nos da una indicación de nuestra fatiga durante los periodos de actividad y de recuperación. La liberación de adrenalina por la tensión del combate eleva los niveles de estrés físico y mental, que impactan negativamente en el proceso de recuperación del atleta [24]. Estos niveles de estrés, además, están relacionados con una alta frecuencia cardíaca por un proceso controlado por el sistema nervioso simpático [25].

En primer lugar, será interesante analizar cuál es el valor óptimo de frecuencia cardíaca media a la que el usuario ha de mantenerse durante el entrenamiento. Para ello, introducimos la **Frecuencia Cardíaca de Reserva oHRR**. La HRR corresponde a la diferencia entre la frecuencia cardíaca máxima y la frecuencia cardíaca basal o en reposo, y permite establecer la **Zona de Frecuencia Cardíaca Óptima (ZFCO)** a la que el usuario ha de entrenar. Para calcular esta zona se sigue el siguiente procedimiento, conocido como fórmula de Karvonen [26].

1. Medición (o estimación) de la frecuencia cardíaca máxima.

Se calcula como el máximo entre la frecuencia cardíaca máxima registrada durante la semana de entrenamiento y la estimada mediante la fórmula de Tanaka (más precisa que la de Karvonen para deportistas) [26]:

$$FC_{maxTanaka} = 208 - 0.7 \cdot Edad \quad (2.1)$$

$$FC_{max} = max(FC_{maxExp}, FC_{maxTanaka}) \quad (2.2)$$

2. Medición de la frecuencia cardíaca basal o en reposo del usuario.
3. Calculamos la HRR como la diferencia entre la FC máxima y la FC basal.

4. Se calculan los límites de la zona de FC objetivo tal que:

$$0.7 \cdot HRR + FC_{Basal} \quad (2.3)$$

$$0.85 \cdot HRR + FC_{Basal} \quad (2.4)$$

Por otro lado, la literatura de los últimos años nos ofrece un conjunto de valores de frecuencia cardíaca por tipo de asalto (actividad o recuperación) de boxeadores amateur de distintos niveles y edades. En la tabla 2.2 [1] se recogen los valores de frecuencia cardíaca absoluta (ppm) recogidos en distintos estudios sobre el impacto de la frecuencia cardíaca en el boxeo [23] [27] [28] [29] que tenían como objeto a boxeadores amateur (femenino y/o masculino) de distinto nivel de entrenamiento (principiante, muy entrenado, competición amateur y élite) en un ámbito de estudio de sparring o simulación de combate. En algunos casos, se enfatiza en la diferencia de resultados a lo largo de los 3 asaltos del combate (*en la tabla, se señala si es asalto o round como Ri, y si asalto de descanso como Di, donde i es el número del asalto en cuestión*) para hacer un análisis del impacto de la fatiga en el número de pulsaciones por minuto obtenido.

Fuente	Nivel Competitivo	Edad	Género	n	HR Absoluto Medio (ppm)	HR Absoluto Máx/ (Mín Descansos) (ppm)
[27]	Principiante	23.7 ± 4.1	Masculino	10	R1 166 ± 19 D1 153 ± 21 R2 173 ± 12 D2 170 ± 19 R3 174 ± 13	R1 180 D1 138 R2 182 D2 146 R3 183
[28]	Muy Entrenado	17.9 ± 1.8	Masculino y Femenino	10	R1 175 ± 8  R2 183 ± 6 R3 186 ± 7	R1 190 ± 8  R2 194 ± 6 R3 199 ± 5
[29]	Amateur	21.4 ± 6.3	Masculino	15	R1 177 ± 12 R2 184 ± 7 R3 187 ± 7	R1 190 R2 192 R3 187
[23]	Élite	21.4 ± 3.0	NE	6	190 ± 7	

Cuadro 2.2: FC absoluta media, máxima y mínima por tipo de asalto según nivel competitivo, edad y género [1].

Los resultados obtenidos reflejan distintos factores. En primer lugar, podemos observar cómo la fatiga acumulada hace que las pulsaciones de los atletas aumenten conforme avanzan los asaltos, tanto de actividad como de recuperación. Por otro lado, pese a la a priori superioridad cardiovascular de los atletas más entrenados, estos reflejan valores de frecuencia cardíaca superiores a los de boxeadores novatos, probablemente por la menor intensidad de entrenamiento de los últimos.

Finalmente, a partir del conjunto de datos analizado, consideramos los rangos de frecuencia cardíaca media en simulación de combate de 171 ppm y 181.3 ppm para los intervalos de actividad de peleadores principiantes y muy entrenados, respectivamente, y de 161.5 ppm para los intervalos de descanso de boxeadores principiantes [27] [28].

Debemos tener en cuenta la posibilidad de que los valores de referencia de la literatura para la FC en los asaltos sean menores que los límites superiores de las zonas de FC

cardíaca óptima (ZFCO). En este caso, no tendría sentido usar dichas referencias, por los que nos quedaremos con el máximo entre ambos.

Recogemos estos valores de referencia en la siguiente tabla:

Nivel de Boxeo	Parámetro de FC	Valor de Ref (bpm)
Principiante	FC Media Asaltos	máx(171, $ZFCO_{85}$ )
	FC Media Descanso	161.5
Muy Entrenado	FC Media Asaltos	máx(181.3, $ZFCO_{85}$ )
	FC Media Descanso	NE

Cuadro 2.3: Valores de referencia de FC media por tipo de asalto

Finalmente, también registraremos los valores pico de esta magnitud y su avance con el paso de las sesiones, ya que esta disminuye al mejorar la capacidad cardiovascular del usuario [30].

Por otro lado, valores bajos de la **Variabilidad de la Frecuencia Cardíaca o HRV**, definida como la variación temporal entre distintas ondas RR originadas en cada latido cardíaco durante un periodo determinado, están relacionados con la mala adaptación, asimilación o recuperación del organismo a la actividad física [31], por lo que su medición nos servirá para calificar esta recuperación y la calidad de nuestro sueño, a su vez indicándonos si nuestro cuerpo está en condiciones de realizar un entrenamiento de cierta intensidad ese día, o si es mejor descansar [32] [33] [34] [35], a modo de predicción.

La literatura aborda el impacto de la recuperación y el estrés de artistas marciales en su rendimiento deportivo mediante el estudio de la variabilidad de su frecuencia cardíaca. Perfiles de mayor nivel de entrenamiento presentan valores más altos de HRV que los menos entrenados [36]. Además, el nivel de carga de entrenamiento está relacionado con la diferencia de HRV pre-post actividad [36], lo que señala el impacto negativo de la sobrecarga de sesiones de entrenamiento en el rendimiento del deportista.

Calificaremos este parámetro mediante su RMSSD, o la raíz cuadrada del valor medio del cuadrado de la diferencia entre ondas RR sucesivas, medido en milisegundos. Estudios reflejan que (dependiendo de las características del usuario) perfiles deportistas rondarán un HRV RMSSD entre 65 y 90 ms [25] [37] (que usaremos como referencia), pero cada usuario tiene su rango basal, medido en reposo.

- **Batería corporal.** El parámetro Body Battery (BB) [38], ideado por Garmin, tiene como objetivo el control de la energía corporal del usuario de manera diaria.

Se basa en un estudio conjunto de la frecuencia cardíaca, su variabilidad (HRV) y el movimiento del usuario, para detectar estados fisiológicos identificables (vigilia/sueño, niveles de actividad y estrés) con objetivo de analizar su impacto en la energía del cuerpo. Este impacto refleja la influencia combinada del estrés, la recuperación y la potencia restauradora del sueño.

A niveles prácticos, expone al usuario de una manera sencilla la predicción de su nivel actividad ese día a partir de los valores de HRV que comentábamos en el apartado anterior.

Además, guarda una relación directa con la forma física de los usuarios, reflejada en el valor máximo del  $VO_2$ . Con una mayor capacidad cardio-respiratoria, la batería corporal se vacía en menor medida, lo que procede de la afirmación de que mejorar la forma física permite gestionar de mejor manera el estrés.

Este parámetro se recarga por la noche (dominio parasimpático del Sistema Nervioso Autónomo (SNA), en niveles de estrés entre 0 y 25, como se explica a continuación), por

lo que encontraremos su valor máximo, idealmente 100, nada más despertarnos, disminuyendo a lo largo del día.

- **Balance calórico.** Debido a la categorización por rangos de peso corporal de la competición de todos los deportes de contacto, el atleta necesitará hacer seguimiento de su ingesta y consumo calóricos con razón de llegar al peso establecido, siendo necesario un riguroso conteo de las calorías gastadas durante las sesiones de entrenamiento. El CMB establece las siguientes categorías por peso corporal para sus competiciones amateur masculinas y femeninas [7].

Divisiones de Peso	Inicia con Kg.	Hasta Kg.
Mosca Ligero	46	48
Mosca	48.1	51
Gallo	51.1	54
Pluma	54.1	57
Ligero	57.1	60
Welter Ligero	60.1	63
Welter	63.1	66
Súper Welter	66.1	69
Medio Ligero	69.1	72
Medio	72.1	76
Súper medio	76.1	80
Semicompleto	80.1	85
Crucero	85.1	90
Completo	90.1	95
Súper-completo	95.1	A+

Cuadro 2.4: Categorías de peso amateur según el reglamento WBC.

- **Frecuencia respiratoria.** Para evitar tener que actuar bajo fatiga durante los asaltos de actividad, lo que impide rendir al máximo de la capacidad física y mental, es indispensable que el atleta se recupere lo máximo posible durante los periodos de descanso intra-round, así como que dosifique sus esfuerzos para cansarse mínimamente en los de actividad. Para ello, una respiración controlada es indispensable, intentando evitar arritmias o respiraciones ventilatorias forzadas que llevan más gasto energético [39].

Por tanto, consideraremos la FR como uno de los principales indicadores de recuperación en el entrenamiento, junto con la FC media durante los intervalos recuperación.

- **Nivel de estrés.** Los deportes de combate producen una reacción hormonal en el organismo, como el aumento de cortisol (hormona del estrés) y de adrenalina [40], que va de la mano con el aumento de la frecuencia cardíaca y respiratoria, produciendo un crecimiento en los niveles de estrés durante los intervalos de actividad [41].

Esto hace interesante la monitorización de este parámetro durante el entrenamiento, ya que podremos ver cómo evoluciona conforme nos fatigamos, o durante los intervalos de recuperación, comprobando que este disminuye.

- **Calidad del sueño.** El usuario dormirá con su dispositivo puesto, ya que éste realiza una estimación de la calidad del sueño del atleta.

Durante el periodo de sueño, en el organismo se producen fenómenos recuperativos, tanto a nivel mental como muscular, y una mala calidad del mismo tiene efecto parámetros cruciales para el boxeo, como la toma de decisiones y ejecución, reflejos, resistencia o fuerza [42].

En cuanto a la recuperación muscular, un mal descanso disminuye la energía reservada para la contracción y la recuperación ante daños musculares [43]. Durante el sueño hay procesos de producción de hormonas como la testosterona y la del crecimiento, cruciales para la recuperación [44].

Por otro lado, estudios muestran la relación entre la duración del sueño con el índice de Masa Corporal, debido a procesos de producción de hormonas relacionadas con el hambre que se producen durante este periodo [44], aumentando el IMC con la reducción de la duración del sueño. Este hecho aumenta el foco en el control de este parámetro, debido a la clasificación de categorías por pesos en las que se basa el deporte.

### Cómo los dispositivos wearables miden estos parámetros

En esta sección se describen los procesos de obtención de los parámetros de interés mediante distintos dispositivos wearables, así como los sistemas de calificación que emplea Garmin para la clasificación del nivel atlético y la consecuente personalización del plan de entrenamiento de sus usuarios.

En primer lugar, tenemos que hacer una diferenciación entre los procesos de captación directa, referente a la obtención del dato a través de los sensores del dispositivo, y los de captación indirecta, en los que incluimos a sistemas y algoritmos que analizan de manera conjunta los parámetros captados de manera directa para obtener otra variable.

Para la obtención de parámetros de manera indirecta, Garmin cuenta con su sistema **First-beat Analytics**, que proporciona analíticas de rendimiento avanzadas a partir de los datos captados por sus sensores, para el ejercicio, la recuperación y el estrés. A pesar de que la compañía brinda la información sobre los datos a partir de los cuales se calculan dichas variables, las fórmulas y algoritmos utilizados no son públicos.

#### ▪ **VO<sub>2</sub> máximo.**

El valor de este parámetro se realiza de manera indirecta. Garmin proporciona al usuario el valor de su VO<sub>2</sub> máximo a partir de una prueba de carrera (recomendablemente de al menos 15 minutos) o de una prueba de ciclismo. Este dato se introduce dentro del perfil de usuario y no se cambia hasta obtener un valor mayor tras repetir una de estas pruebas.

Para determinar este valor, realiza una estimación del VO<sub>2</sub> máximo con su sistema First-beat Analytics identificando, analizando e interpretando los datos de rendimiento de la carrera, como la relación entre el ritmo de carrera y el esfuerzo ejercido por el cuerpo, que en la mayoría de los casos es lineal.

Se clasifica en 5 niveles, según el sexo y la edad del usuario, como se muestra en la siguiente tabla:

#### ▪ **y HRV**

Los valores de frecuencia cardíaca y sus derivados se obtienen principalmente a partir de dos sensores diferentes:

- Los dispositivos wearables de muñeca, como los relojes y las pulseras inteligentes, realizan esta medición mediante un sistema óptico, conocido como OHR, que consiste en la emisión de luz verde a través de los vasos sanguíneos, donde los glóbulos rojos viajan en un flujo por pulsos a la frecuencia cardíaca. El sensor óptico mide la variación de este flujo detectando la intensidad de la luz reflejada.
- Por su parte, los sensores específicos de frecuencia cardíaca, equipados generalmente cerca del corazón (en una banda pectoral, por ejemplo), utilizan un sistema de

Nivel	Percentil	20–29 H/M	30–39 H/M	40–49 H/M	50–59 H/M	60–69 H/M	70–79 H/M
<b>Superior</b>	95	55,4 /49,6	54 /47,4	52,5 /45,3	48,9 /41,1	45,7 /37,8	42,1 /36,7
<b>Excelente</b>	80	51,1 /43,9	48,3 /42,4	46,4 /39,7	43,4 /36,7	39,5 /33	36,7 /30,9
<b>Bueno</b>	60	45,4 /39,5	44 /37,8	42,4 /36,3	39,2 /33	35,5 /30	32,3 /28,1
<b>Aceptable</b>	40	41,7 /36,1	40,5 /34,4	38,5 /33	35,6 /30,1	32,3 /27,5	29,4 /25,9
<b>Deficiente</b>	0–40	41,7 /36,1	40,5 /34,4	38,5 /33	35,6 /30,1	32,3 /27,5	29,4 /25,9

Cuadro 2.5: Niveles Garmin de VO<sub>2</sub> Máximo según sexo y edad.

electrocardiograma (ECG) consistente en la detección de las señales eléctricas que el corazón emite para activar la contracción de sus músculos.

Por otro lado, Garmin hace un seguimiento de la variabilidad de la frecuencia cardíaca mediante el sensor de luz verde que es reflejada con cada latido. Esta medición se hace de manera continua durante el periodo de sueño [45] y se evaluará con un sistema de cuatro niveles adaptado a cada usuario al determinar un valor basal típico según los valores que registre durante un periodo 'inicial'. Estos niveles son: balanceado (si está dentro del rango basal o típico), desbalanceado superior (si se pasa del límite superior), desbalanceado inferior (si se obtiene un valor inferior al rango basal en 10 ms o menos) y bajo (si el valor obtenido es menor que el rango basal con una diferencia mayor de 10 ms).

El inconveniente es que el Garmin Venu 2 no cuenta con este sistema de calificación del HRV, por lo que tendremos que medir de manera propia el valor basal de cada usuario.

#### ■ Batería corporal

Este parámetro desarrollado por Garmin se obtiene de nuevo con su sistema Firstbeat Analytics, a partir de la frecuencia cardíaca, su variabilidad (captados con el sensor óptico del dispositivo) y el movimiento del usuario (medido con el acelerómetro).

Garmin categoriza el nivel de este parámetro a lo largo del día con un sistema de cuatro niveles puntuado del 5 al 100:

Calificación	Valor
Excelente	$\geq 75$
Buena	$\geq 50$
Regular	$\geq 25$
Muy baja	$< 25$

Cuadro 2.6: Categorización de Garmin del Body Battery según su valor

#### ■ Balance calórico.

Para calcular el gasto calórico del usuario con wearables, se realiza un estudio del nivel de actividad (calculado con datos de movimiento, registrados por el acelerómetro, y de frecuencia cardíaca), en conjunto a la información del perfil de usuario, como su edad, peso y sexo.

Garmin diferencia entre calorías BMR (o en reposo), definidas como el gasto calórico equivalente a la energía que el cuerpo necesita para mantener sus funciones vitales cuando

está en reposo, y las activas o consumidas cuando el cuerpo se encuentra realizando una actividad.

### ■ Frecuencia respiratoria.

Garmin calcula la frecuencia respiratoria con su sistema Firstbeat Analytics, basándose en una relación entre este parámetro y el HRV.

Esta relación tiene que ver con un fenómeno biológico conocido como Arritmia Sinusal Respiratoria ASR, que consiste en que el tiempo entre pulsaciones consecutivas se acorta o se alarga según se inhale o se exhale, respectivamente [46].

### ■ Nivel de estrés.

La medición del nivel de estrés se basa en los estudios sobre el modo de dominio del SNA. El dominio parasimpático del SNA se relaciona con los períodos de relajación del usuario, mientras que cuando éste se encuentra acelerado y agitado, o en una situación de estrés, el dominio del SNA es simpático.

Para medir este parámetro, se utilizan algoritmos a partir de los valores de la frecuencia cardíaca y su variabilidad.

Garmin mide el nivel de estrés en una escala de 0 a 100, diferenciando en una zona de 0 a 25, en la que el cuerpo se encuentra en un estado de reposo asociado al dominio parasimpático del sistema nervioso autónomo (se recarga el Body Battery), y de 25 a 100 en la que el dominio del SNA es simpático, referente a situaciones de estrés (se gasta el Body Battery). Además, categoriza este valor en cuatro niveles dentro de la escala:

Calificación	Valor
Estrés alto	$\geq 75$
Estrés medio	$\geq 50$
Estrés bajo	$\geq 25$
Estado de reposo	$< 25$

Cuadro 2.7: Categorización de Garmin del nivel de estrés según su valor

### ■ Calidad del sueño.

Los wearables del mercado miden la calidad del sueño de los usuarios a partir de los datos brindados por su sensor óptico, como la frecuencia cardíaca y su variabilidad, que brinda, como hemos comentado, información sobre la frecuencia respiratoria. Además, su acelerómetro permite determinar cuándo el sujeto se duerme y se levanta.

El sueño se clasifica en tres etapas:

- Sueño ligero. La intensidad de los movimientos oculares y la actividad muscular empieza a disminuir.
- Sueño profundo. Cesan los movimientos oculares y la actividad muscular. Disminuyen las frecuencias cardíaca y respiratoria.
- Sueño REM. Etapa en la que la actividad cerebral está al mismo nivel que en el estado de vigilia.

Garmin ofrece un sistema de puntuación del sueño en una escala de 0 a 100, basándose en las 3 etapas anteriores y en la duración del mismo, dando insights con los factores que han influido en esta puntuación. Además, tiene en cuenta factores como el estrés, la inquietud (medida como el movimiento durante este periodo) y el despertar (según la rapidez del cambio de pulsaciones).

### Cuándo medir estos parámetros

En esta sección se analiza cuándo es más conveniente medir los parámetros comentados, según el criterio que nos brinda la literatura de la ciencia del deporte, para poder diseñar el plan de monitorización del rendimiento y la recuperación del usuario más óptimo.

- **VO<sub>2</sub> máximo.** La cantidad de oxígeno máxima consumida por unidad de tiempo es un valor que puede aumentar de la mano del rendimiento cardiovascular, lo que significa que al hacernos más fuertes y eficientes nuestro valor de VO<sub>2</sub> Máximo aumentará. Esto augura que la medida inicial de este parámetro no siempre será el valor máximo que nuestro VO<sub>2</sub> puede alcanzar, ya que, la práctica de ejercicio con constancia en un periodo de tiempo puede resultar en un crecimiento de este valor.

Por tanto, comenzaremos con una medición inicial de nuestro VO<sub>2</sub> máximo con nuestro dispositivo wearable, con una prueba de carrera de al menos 15 minutos. Repetiremos este test una vez semanalmente, o nuestro caso tras la semana de entrenamiento, para poder analizar la evolución de la capacidad cardiovascular del atleta, a lo largo de las semanas con el entrenamiento planeado.

La literatura incide sobre la mayor capacidad cardiorrespiratoria de boxeadores de mayor nivel, y de boxeadores senior frente a los de edades más tempranas [21] [22] [47], lo que respalda la medición periódica del valor de VO<sub>2</sub> máximo a partir de la inicial, que como se ha comentado no siempre supone el valor máximo real.

- **HR y HRV.** Monitorizaremos la frecuencia cardíaca durante el transcurso del entrenamiento de boxeo, para poder relacionarla, como se ha comentado, con la fatiga conforme avanzan los asaltos.

En cuanto a la variabilidad de la frecuencia cardíaca, será interesante registrarla a primera hora de la mañana, ya que esta medición refleja cómo ha asimilado nuestro cuerpo las actividades y circunstancias (como el estrés) del día anterior, y como se ha recuperado ante estas, además de da una indicación de la calidad de nuestro sueño [48].

Expertos subrayan la importancia de hacer la medición recién despiertos [28] (frente a de manera al entrenamiento) ya que diversos estudios [32] [33] evidencian la alteración de este valor por factores como las comidas, viajes y estímulos a lo largo del día. Aparte, permite planificar con cierto margen el entrenamiento del día, ya que sirve de predicción de la capacidad de nuestro organismo de afrontar la actividad o si, por el contrario, es más apropiado mantener otro día de descanso [34].

Un HRV por debajo del umbral indicará una mala recuperación del cuerpo [31].

- **Batería corporal.** Seguiremos la cuenta regresiva de este parámetro a lo largo del día. Esto nos dará información sobre el nivel de energía con el que partimos al levantarnos, el balance energético registrado al medir este parámetro antes y después del entrenamiento y, finalmente, el gasto energético total registrado al acostarnos.
- **Balance calórico.** Es un parámetro que se controla diariamente, ya que ese gasto calórico tendrá que ser compensado con la ingesta de alimentos necesaria para nuestro objetivo de peso corporal. Debido a la alta intensidad del entrenamiento, controlaremos el gasto calórico durante su transcurso.
- **Frecuencia respiratoria.** Del mismo modo que la frecuencia cardíaca, nos interesa conocer la evolución del número de respiraciones por minuto a lo largo de los asaltos de actividad y de recuperación, esperando un aumento de la misma por el impacto del estrés y de la fatiga.

Debido a las limitaciones de monitorización por la sensibilidad de los sensores frente a movimientos, nos centraremos en la información resultante de los intervalos de recuperación, donde el usuario se encuentra en reposo.

- **Nivel de estrés.** Nos interesa monitorizarlo durante todo el día debido a su efecto en la recuperación y rendimiento físico, pero también como cambia durante los propios períodos de actividad, ya que, como comentábamos, afectará a la recuperación intra-round o, dicho de otra forma, a nuestra eficiencia cardiovascular (por el aumento de pulsaciones).

El inconveniente es que Garmin no mide este parámetro durante la actividad física, ya que la tensión implícita en la misma puede confundirse con los niveles de estrés del usuario. Por tanto, nos quedaremos con los datos registrados en las fases ajenas al entrenamiento.

- **Calidad de sueño.** La calidad del sueño se medirá con nuestro dispositivo wearable mientras dormimos. Junto al HRV nos dará información necesaria para analizar cómo ha asimilado el organismo el entrenamiento y cómo se ha recuperado ante el mismo.



### 3. Materiales y métodos

#### 3.1. Hardware empleado

En esta sección comentaremos el equipo utilizado, desde sus características hasta su configuración, para la fase de captura de los datos de análisis.

##### 3.1.1. Reloj inteligente Garmin Venu 2

El segundo modelo de la gama fitness por excelencia de Garmin, cuenta con la tecnología necesaria para hacer seguimiento del rendimiento deportivo y de parámetros generales de salud. Permite la monitorización de parámetros como la frecuencia cardíaca, la frecuencia respiratoria y su variabilidad, el estrés, el sueño o la pulsioximetría, así como del estado del cuerpo con el análisis combinado de varios parámetros, como la batería corporal o ‘Body Battery’.

Cuenta con aplicaciones propias para la medición de parámetros de salud y la realización y monitorización de actividades deportivas, además de, como hemos comentado, permitir la descarga de aplicaciones de terceros.



Figura 3.1: Smartwatch Garmin Venu 2

### Sensores

Para la medición y monitorización de parámetros fisiológicos y de salud, este reloj cuenta con una serie de sensores integrados:

- GPS, Glonass y Galileo. Sensores de posicionamiento global, diferenciados entre sí en el país que los desarrolló y los gestiona (EEUU, Rusia y Europa).
- Monitor de frecuencia cardíaca Garmin Elevate. Sensor óptico (por PPG o fotopletismografía) de frecuencia cardíaca, diseñado para monitorizar la FC de manera ininterrumpida 24/7. Situado visiblemente en la parte posterior del reloj, emite luz verde hacia la piel, captando la reflexión en los glóbulos rojos de la sangre en el flujo establecido por las contracciones del corazón a la frecuencia cardíaca [49].
- Pulsioxímetro. Para medir la cantidad de oxígeno en sangre o la saturación de oxígeno ( $\text{SpO}_2\%$ ), Garmin emplea una combinación de luces rojas e infrarrojas hacia el torrente sanguíneo, permitiendo medir, por ejemplo, como el cuerpo se está ‘aclimatando’ a cambios de altitud [50].
- Altímetro barométrico y brújula. El altímetro determina la altitud a la que se encuentra el dispositivo (respecto al nivel del mar) midiendo la presión atmosférica en el aire, según la relación inversa entre ambas magnitudes. La brújula permite conocer la posición hacia la que se orienta el usuario.
- Giroscopio y acelerómetro. Sensores de movimiento del reloj. El primero detecta la velocidad de rotación respecto a un eje determinado y los ángulos de inclinación. Por su parte, el acelerómetro mide cambios de velocidad en una dirección determinada, mediante un sistema de capacitancia por la separación ligada al cambio de movimiento de dos estructuras semiconductoras.
- Termómetro. Para medir la temperatura ambiental o externa.
- Sensor de luz ambiental. Con el objetivo de optimizar la batería cambiando automáticamente la intensidad de brillo de la pantalla.

### Características físicas del dispositivo

En esta sección analizamos los materiales y características de los que están fabricados los componentes que estructuran el reloj, tales como pantalla, carcasa, bisel o correa. Esto es crucial para nuestro estudio, ya que nuestro wearable ha de ser tanto liviano y cómodo como resistente para cumplir con los requisitos físicos comentados en la introducción de este capítulo. En la siguiente tabla se recogen dichas propiedades y características:

Estos materiales son los más resistentes de entre los relojes inteligentes de la misma gama de precio del mercado, destacando el material de la pantalla, el elemento más sensible a golpes del reloj, fabricado con el cristal Gorilla Glass 3, fortalecido químicamente y diseñado con una composición de vidrio patentada con gran capacidad para resistir rayones y golpes, siendo hasta 4 veces más resistente que sus competidores. Además de la resistencia de los propios materiales, añadimos un protector de pantalla extra de silicona.

En cuanto al hardware correspondiente a su funcionamiento, el reloj cuenta con los siguientes botones, representados en la figura 3.2:

- A. Botón de acción (Select). Al mantenerlo pulsado con el reloj apagado, se encenderá. Al pulsarlo con el reloj encendido, accederemos al menú de actividades. Al pulsarlo durante una

Características	Descripción
Material de la correa	Silicona
Material de la lente	Corning Gorilla Glass 3
Material del bisel	Acero inoxidable
Material de la carcasa	Polímero reforzado con fibra
Tamaño de la carcasa (An. x L. x Al.)	45,40 x 45,40 x 12,20 mm
Peso	49 g
Tamaño de la pantalla (Ancho/Alto)	Diámetro de 1,30" (33 mm)
Resolución de pantalla (Ancho/Alto)	416 x 416 píxeles

Cuadro 3.1: Materiales y características de los componentes físicos del Garmin Venu 2

actividad, la pausaremos y reanudaremos. Al mantenerlo pulsado con el reloj encendido, accederemos al menú de controles.

- B. Botón atrás (Back). Con el reloj encendido, pulsarlo devuelve la pantalla anterior. Mantenerlo pulsado dirige al menú de configuración. Pulsarlo durante una actividad cambia la fase de la misma.
- C. Pantalla táctil. Pulsando la pantalla podemos hacer selecciones. Pulsando dos veces activa el reloj. Deslizando hacia arriba o abajo accedemos a los widgets. Deslizando hacia la derecha hacemos un Back.



Figura 3.2: Esquema de botones del Garmin Venu 2

## Autonomía

La duración de la batería del dispositivo es otro factor a tener en cuenta para su elección, ya que necesitaremos un funcionamiento autónomo lo suficientemente longevo para una monitorización continua durante una semana.

El fabricante tabula la duración de la batería del reloj según sus modos de uso, teniendo en cuenta que estos tiempos se acortarán con el uso continuado de ciertos sensores, siendo el tiempo de uso real más reducido:

Concretamente, activando el modo de medición por segundo con el que trabajamos en este estudio, la duración de la batería se reduce a unos 5 días. En el apartado de resultados haremos

Modo	Duración
Modo smartwatch	Hasta 11 días
Modo GPS con música	Hasta 8 horas
Modo GPS sin música	Hasta 22 horas
Modo smartwatch de ahorro de energía	Hasta 12 días

Cuadro 3.2: Autonomía del Garmin Venu 2 según el modo de uso

un inciso sobre la duración real de la autonomía en este modo de medición de datos.

### Garmin Connect

A continuación, se introduce la plataforma de gestión y almacenamiento de datos de Garmin, sobre la que nos apoyaremos en este proyecto, así como su entorno de desarrollo de aplicaciones.

Garmin es la empresa de dispositivos wearables por excelencia. No es solo la pionera en sus dispositivos como producto final, sino que ha desarrollado la tecnología y los protocolos estandarizados del sector, adoptados por el resto de compañías, como es el caso de Polar. Además, ha desarrollado parámetros y variables propias, a partir de estudios y análisis sobre parámetros primordiales (como la frecuencia cardíaca, frecuencia respiratoria, etc.), como es el caso del Body Battery, fundamentados científicamente y soportados por el propio peso de la marca.

Pese a que la compañía ofrece su propia interfaz de usuario y aplicaciones para la monitorización de diversas actividades deportivas, permite la personalización de la experiencia de uso del dispositivo. Desde el desarrollo de aplicaciones que se ajusten mejor a las necesidades del usuario al realizar dichas actividades, hasta el uso de para la visualización y el análisis de los datos registrados, la flexibilidad en el uso de los dispositivos es uno de los diferenciadores de Garmin.

Esta plataforma se sujeta en los siguientes pilares:

- **Garmin Connect.** Plataforma de almacenamiento, análisis y visualización de datos diseñada por Garmin para mejorar la experiencia de monitorización de parámetros de los usuarios. Permite la visualización de la información detallada registrada durante un determinado periodo de tiempo, tanto de las actividades físicas realizadas como los parámetros de salud capturados. La encontramos como aplicación móvil o de escritorio, realizándose la sincronización entre el dispositivo externo y el wearable sobre Wifi o por cable, respectivamente.
- **Almacenamiento en la nube.** Garmin Connect cuenta con un sistema de almacenamiento en la nube a la que los usuarios pueden solicitar acceso uniéndose al programa de desarrollador de Garmin. Desde estas bases de datos podremos descargar ficheros JSON o FIT con los datos obtenidos en la monitorización cada segundo.
- **Connect IQ SDK.** Entorno de desarrollo de Garmin, con las herramientas necesarias para programar y simular aplicaciones para sus dispositivos. Se trabaja en el editor de código Visual Studio Code, descargando la extensión de Monkey C.

Monkey C es el lenguaje de programación diseñado específicamente para el desarrollo de aplicaciones Garmin. Con una sintaxis parecida a Java o C#, tiene en cuenta las contrariedades y limitaciones de desarrollo como consecuencia de las pantallas (a veces) circulares de pequeño tamaño con baja capacidad gráfica y de resolución, recursos limitados de

memoria y almacenamiento, limitaciones de tiempo de ejecución o restricciones de libre acceso a datos. El proceso de puesta a punto del entorno, así como el tutorial de creación de un programa, queda explicado en la página de Garmin Developers [51].

Lo más destacable del sistema de programación de Garmin Connect IQ es el uso de APIs para el manejo de datos. Garmin proporciona información sobre sus APIs, el método principal de acceder a los datos capturados y exportarlos. Tendremos una API para cada sensor o parámetro, con funciones específicas, por lo que una de las características de las APIs de Connect IQ es que tienen que estar hechas ‘a medida’ para cada uno de los dispositivos Garmin, debido a que cuentan con sensores específicos, con APIs concretas. Esto significa que nosotros, como desarrolladores, no tenemos que preocuparnos de calcular los datos, ya que ese proceso se realiza en el propio controlador del dispositivo. Esto es una de las características principales de este tipo de dispositivos, que nos simplifica mucho la programación, como veremos en el apartado de programación.

- **Connect IQ Store** Plataforma en la que los usuarios, tras un proceso de análisis y chequeo de Garmin, podrán subir sus aplicaciones para ser descargadas, así como descargarlas de forma gratuita o de pago.
- **ANT y ANT+.** Para la comunicación entre dispositivos en redes personales en los que se despliegan varios, Garmin desarrolló ANT, un protocolo inalámbrico de transferencia de datos por el aire sobre Bluetooth de bajo consumo (BLE). Podemos trabajar con este protocolo de manera sencilla con ANT+, un set de perfiles de acuerdos de comunicación entre dispositivos, que trabaja por encima de ANT.

Es la estandarización de los mensajes mandados entre dispositivos en la industria wearable, usada no solo en los dispositivos de la compañía, sino en muchos de los productos de la competencia.

### 3.1.2. Sensor de frecuencia cardíaca Garmin HRM-Pro Plus.

Como hemos comentado, necesitamos un sensor que complemente al smartwatch, superando las limitaciones de los sensores del mismo impuestas por el movimiento de los brazos durante el entrenamiento.

Garmin ha desarrollado un modelo de sensor de frecuencia cardíaca capaz de competir con el gold estándar del sector, la Polar H10, no solo por la precisión brindada por su tecnología de medición por electrocardiograma (ECG), frente a la monitorización óptica (OHR) con sensores de luz verde implementada en la mayoría de relojes inteligentes, sino por contar con el respaldo y el soporte que ofrece la infraestructura de la compañía número uno en wearables.

Para transmitir información en tiempo real al dispositivo al que está conectado, cuenta con los protocolos ANT+ y BLE. Esta información no solo es referente a la frecuencia cardíaca y su variabilidad, sino también parámetros de actividad independiente, como pasos, calorías o minutos de intensidad. Estos datos pueden transmitirse IRT o ser almacenados en una memoria temporal para ser enviados al finalizar el entrenamiento.



Figura 3.3: Sensor de frecuencia cardíaca Garmin HRM-Pro Plus.

## ECG

El sensor de frecuencia cardíaca Garmin HRM-Pro Plus funciona por electrocardiograma, a diferencia de los wearables de muñeca, como smartwatches o pulseras inteligentes.

Nuestro corazón emite señales eléctricas para activar la contracción de sus cuatro cámaras para el bombeo de sangre al ritmo de la frecuencia cardíaca. La tecnología ECG consiste en colocar sensores con electrodos para medir los cambios de las señales eléctricas emitidas por el corazón y transmitidas por los fluidos del organismo hasta la piel, permitiendo realizar una representación gráfica de las mismas, conocida como electrocardiograma [52].

### Características físicas del dispositivo

Los requerimientos físicos de movimiento de los deportes de contacto comentados al inicio del capítulo suponen la necesidad de utilizar un sensor que funcione con buena precisión de manera independiente al movimiento, por lo que una banda pectoral es ideal para nuestra monitorización.

La siguiente tabla recoge las características físicas del dispositivo, así como las referentes su autonomía [53]:

Como podemos comprobar, sus reducidas dimensiones y su ligero peso hacen que el sensor sea cómodo para el usuario. Además, que la correa sea ajustable permite que el sensor quede totalmente pegado e inmóvil durante el entrenamiento. Es resistente al agua, por lo que el sudor no supone un problema para la medición.

#### 3.1.3. Arquitectura de la red personal

Una vez conocidos los dispositivos wearables que emplearemos para la captura de los datos de los usuarios durante la semana de actividad, vamos a terminar de definir la arquitectura del proyecto, detallando la composición de la capa de captación de la arquitectura por capas de un sistema de monitorización con dispositivos wearables, expuesta en la sección 2.1.2.

Esta arquitectura se basa en un sistema formado por un dispositivo pasarela, conectado a la infraestructura de gestión y almacenamiento de los datos, que estará conectado, a su vez, al

Características	Descripción
Dimensiones (con correa) (Ancho/Alto/Profundidad)	60,0-142,0 x 3,0 x 1,2 cm
Dimensiones del sensor (Ancho/Alto/Profundidad)	29,4 x 51,4 x 8,5 mm.
Medidas correa	60-106 cm Con extensor de correa opcional: 60-142 cm
Peso	52 g
Clasificación de resistencia al agua	Nadando, 5 ATM
Batería	CR2032
Autonomía de la pila/batería	12 meses
Tipo de tapa de la pila	Sin herramientas
Material del sensor	Plástico

Cuadro 3.3: Características físicas del Garmin HRM-Pro Plus.

sensor externo que le brindará información concreta (en este caso de frecuencia cardíaca) para obtener una mayor precisión.

- **Dispositivo pasarela.** En este caso, el smartwatch Venu 2, con acceso a toda la infraestructura de almacenamiento de Garmin Connect.
- **Sensor externo.** La banda pectoral con sensor de frecuencia cardíaca HRM-Pro Plus se conectará al Venu 2 para mejorar la precisión de las medidas de frecuencia cardíaca obtenidas. Esta información se manda al dispositivo pasarela y se gestiona a través de la infraestructura de Garmin.
- **Protocolo de comunicación entre dispositivos.** La comunicación entre ambos dispositivos se realiza sobre el protocolo inalámbrico ANT+, desarrollado por el propio Garmin.
- **Protocolo de comunicación externa.** Para la transferencia de datos entre el dispositivo pasarela y la infraestructura de gestión y almacenamiento, típicamente accedida desde una aplicación móvil o de escritorio, Garmin cuenta con WiFi y transferencia por cable, respectivamente.

Representamos la arquitectura descrita en la siguiente figura:



Figura 3.4: Capa de captación. Arquitectura de la red personal. Elaboración propia.

### 3.1.4. Sincronización de ambos dispositivos

Para que esta comunicación inalámbrica pueda llevarse a cabo, el usuario deberá sincronizar ambos dispositivos cada vez que vaya a realizar una sesión de entrenamiento. Para ello, se han de seguir los siguientes pasos:

1. En el reloj, accederemos al apartado de Ajustes manteniendo pulsado el botón Back.
2. Accederemos a ‘Sensores y accesorios’ -*‘Añadir nueva’*.
3. Nos equiparemos la banda pectoral, que automáticamente encenderá el sensor de frecuencia cardíaca.
4. Con el sensor encendido y situado menos de 3 metros de distancia del reloj, seleccionaremos en este último ‘Frecuencia cardíaca’, comprobando que se está buscando el sensor.
5. Una vez seleccionado el dispositivo, comprobamos que funciona correctamente observando datos de FC en el reloj sin llevarlo puesto.

## 3.2. Software empleado

Esta sección servirá para introducir las plataformas y herramientas principales con las que se ha desarrollado el depurado y el análisis de los datos capturados, así como la implementación del algoritmo propuesto descrito en la sección 3.3. La lista completa de elementos software utilizados para el desarrollo del proyecto se encuentra en el apéndice D.

### 3.2.1. Anaconda Navigator

‘El Portal de Escritorio a la Ciencia de Datos’, como sus desarrolladores definen al navegador Anaconda [54], es la interfaz gráfica de usuario de la distribución de software libre Anaconda, diseñada para el uso de aplicaciones y la gestión de entornos sin la necesidad de una interfaz de comandos. Anaconda se centra en la distribución de los lenguajes Python y R, y en los últimos años ha puesto su foco en el desarrollo de iniciativas de Inteligencia Artificial.

Cuenta con una gran variedad de entornos de código abierto y biblioteca para el minado, análisis y visualización de datos, además de entornos de desarrollo, a los que añade funcionalidades de IA, como asistentes virtuales.

En este proyecto se trabaja con la versión 2.6.2 (Julio 2024) de Anaconda Navigator, con el entorno Jupyter Notebook V 7.2.2.

### Jupyter Notebook 7.2.2

El proyecto Jupyter [55] es una plataforma web interactiva para el desarrollo de software de código abierto, centrado en lenguajes de computación (como vemos en su propio nombre: JULia, PYThon y R).

Desde Anaconda Navigator, trabajamos con notebooks de Jupyter, que nos permite ejecutar fragmentos de código con resultados inmediatos sin necesidad de usar una CLI.

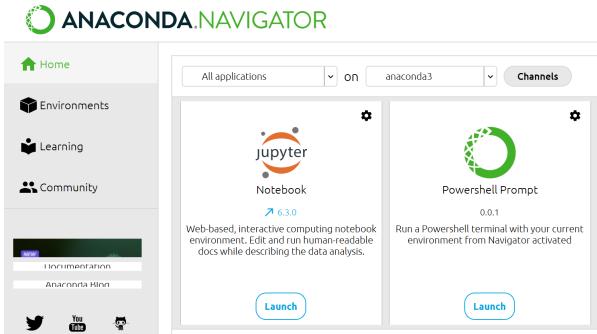


Figura 3.5: Launcher de Jupyter Notebooks en Anaconda Navigator.

En el caso de este proyecto, se usará Python 3 como el lenguaje de desarrollo del programa de análisis de los datos, usando como herramienta las funciones del siguiente conjunto de módulos y librerías, con *Pandas* como la principal para el análisis de datos:

- **os.** El primer módulo nos permite utilizar funcionalidades del sistema operativo en el que se ejecuta el notebook. Lo usaremos para navegar por los directorios en los que se encuentran los datos, listarlos y cargarlos en el script [56].
- **json.** Este módulo permite trabajar con archivos JSON. Se combina con la librería Pandas para trabajar con datos en este formato [57].
- **re.** El módulo re debe su nombre a que se usa para trabajar con expresiones regulares, referentes a patrones para buscar, extraer o manipular texto. Se utilizará para la gestión de directorios [58].
- **matplotlib.pyplot.** Es la biblioteca utilizada para la representación gráfica de los datos [59].
- **find\_peaks de scipy.signal.** El módulo `scipy.signal`, de la librería de ciencia SciPy, contiene funciones para el procesado de señales. Concretamente, `find_peaks` se usará para la detección de máximos y mínimos locales [60].
- **pandas.** Se importa la librería Pandas para la manipulación de datos como DataFrames (DF). Como se ha comentado, se combinará con el módulo json.

### Librería Pandas

Pandas es una herramienta de código abierto sobre Python, para el análisis y la gestión de datos de una manera eficiente, simple e integrable a otras librerías.

Importaremos esta librería desde nuestro notebook para trabajar con nuestros datos convertidos en DataFrames (DF). Podemos definir estos DF como una estructura organizada como tabla de datos con filas y columnas, donde cada columna es una serie de elementos indexados.

Las principales funciones de Pandas en las que nos apoyaremos para el análisis de los datos son las que listamos a continuación [61]:

- **DataFrame.** Nos permite crear un DataFrame a partir de una lista o diccionario de Python.
- **`loc[x, y]`.** Permite acceder y manipular el elemento concreto, situado en la fila  $x$  de la columna  $y$ , de un DF.

- **iloc[i]**. Similar a **loc**, permite acceder a elementos específicos del DataFrame pero utiliza índices de posición en lugar de etiquetas de fila o columna. Esto nos permite acceder al elemento *i* de una columna.
- **apply**. Aplica una función a lo largo de un eje (filas o columnas) del DataFrame, permitiendo realizar transformaciones personalizadas en los datos.
- **concat**. Une o concatena varios DataFrames a lo largo de un eje especificado (filas o columnas).
- **Categorical**. Convierte una columna en un tipo categórico, lo que permite definir un orden específico en los valores de la columna.
- **sort\_values**. Ordena los datos del DataFrame en función de los valores de una o más columnas.
- **dropna**. Elimina las filas o columnas que contienen valores nulos (NaN), ayudando a limpiar el DataFrame.
- **Series**. Crea una serie de Pandas, una estructura unidimensional que puede almacenar cualquier tipo de datos (números, cadenas, etc.).
- **merge(DF1, DF2, on, how)**. Permite aunar DFs en base a una columna común, cuyo nombre se especifica en el parámetro *on*, con el método de unión indicado en *how*.
- **shift(i)**. Este método se usa para desplazar las filas de una columna *i* veces.
- **copy()**. Permite crear una copia del DF en cuestión. Se usa ya que hay métodos, como shift, que no recomiendan aplicarse sobre los DF originales, ya que pueden modificarlo.
- **fillna(i)**. Método que sustituye los valores nulos (NaN y None) de un DF por el valor *i*.

### 3.2.2. Aplicaciones de Garmin utilizadas

En esta sección se describen las aplicaciones y actividades propias de Garmin utilizadas para la medición manual de los datos del estudio.

Además del uso de las aplicaciones con su configuración por defecto, Garmin ofrece la opción de personalización de sus actividades deportivas, permitiendo seleccionar qué parámetros y variables se monitorizarán y mostrarán durante el tránscurso de dicha actividad. Nos apoyaremos de esta funcionalidad para la monitorización del entrenamiento de boxeo.

#### Prueba de Carrera

La prueba de carrera de Garmin es una de las dos maneras que la compañía ofrece para registrar el valor de VO<sub>2</sub> máximo del usuario. Para ello, el usuario deberá realizar una carrera de al menos 15 minutos en el exterior.

Monitoriza métricas de carrera como el ritmo, la distancia, el recorrido y la duración, además de parámetros fisiológicos como la frecuencia cardíaca y el gasto calórico.

Para iniciar un prueba de carrera, el usuario seleccionará el menú de actividades con el botón superior del reloj.

## Resumen de Salud

La prueba Resumen de Salud de Garmin es el medio principal que la compañía nos brinda para registrar nuestro valor de variabilidad de frecuencia cardíaca o HRV.

Consiste en una prueba de dos minutos en la que el usuario tendrá que sentarse y permanecer inmóvil y relajado para que el movimiento no interfiera en la precisión de los sensores.

Además del valor del HRV, nos brinda información como la frecuencia cardíaca, pulsioximetría, estrés y frecuencia respiratoria, haciendo un resumen conjunto del estado cardiovascular del usuario.

Podemos acceder a esta prueba desde el menú de actividades, pulsando el botón Select de nuestro smartwatch.

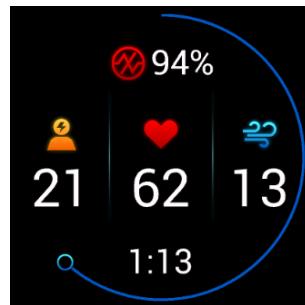


Figura 3.6: Display de la prueba Resumen de Salud en un reloj Garmin.

## Barra

Usaremos la actividad 'Barra' para la monitorización del entrenamiento de boxeo.

Es una actividad personalizada a partir de la aplicación propia de Garmin para yoga, debido a que de su elenco de actividades deportivas, es la única con acceso a los parámetros de nivel del estrés y la frecuencia respiratoria, de gran importancia en nuestro estudio (2.2.3).

De entre las variables que monitoriza, destacamos la **frecuencia cardíaca** (media y máxima), el nivel de **estrés**, la **frecuencia respiratoria**, las **calorías** consumidas y la variación de la **batería corporal**, además de la duración del entrenamiento.

Al ser una personalización de una aplicación propia de Garmin, es compatible a nuestro sensor de frecuencia cardíaca.

Para iniciar esta actividad, accedemos al menú de actividades y deslizaremos hacia abajo hasta el menú de actividades adicionales, en el que se encuentran las descargadas desde la Garmin Store y las personalizadas, como es el caso.



Figura 3.7: Pantalla inicial y display de la actividad Barra en nuestro Venu 2

### 3.3. Algoritmo de calificación de la recuperación y el rendimiento deportivo

En esta sección se describe el algoritmo planteado para relacionar la recuperación previa al entrenamiento con el desempeño en el mismo, mediante un sistema de rúbricas que califica tanto la recuperación como el rendimiento deportivo, según un sistema de rúbrica basado en la caracterización fisiológica de los deportes de contacto llevada a cabo en el Capítulo 2. Hay que recalcar que **este sistema de calificación se basa de una interpretación de la literatura de las ciencias de deportes de contacto revisada, por lo que no cuenta con rigor científico** y necesitaría un proceso de validación científica realizado por especialistas, como se subraya en la sección de conclusiones y líneas futuras, 6.2,

El objetivo es que el usuario disponga de una visualización de sus datos que le permita asociar la calidad de su recuperación con el rendimiento del entrenamiento dicho día. Para ello, se complementará el algoritmo con un sistema de insights con comentarios y recomendaciones según la calificación de los parámetros obtenida.

Recuperación y rendimiento serán evaluados con rúbricas compuestas por distintos parámetros, cada uno de ellos calificados con cuatro variables categóricas que conforman un sistema de niveles análogo al que usa Garmin para sus variables. Cada calificación puntuará del 1 al 4:

Calificación	Puntuación
Excelente	4
Bueno/a	3
Regular	2
Malo/a	1

Cuadro 3.4: Niveles de calificación y su puntuación.

Los parámetros usados para las rúbricas de recuperación y rendimiento y su modo de calificación se describe más adelante.

A cada uno de los parámetros, según su nivel de importancia considerado y basado en una justificación que se planteará en las siguientes sub-secciones, se le asignará un peso. Así, la calificación global diaria ( $C_{Diaría}$ ) de cada uno de los aspectos se calcula como ( $N$ : número de parámetros,  $p_n$ : peso del parámetro  $n$  y  $C_n$ : calificación del parámetro  $n$ ):

$$C_{Diaría} = \frac{\sum_{n=1}^N p_n \cdot C_n}{P} \quad n = 1, \dots, N \quad (3.1)$$

$$P = \sum_{n=1}^N p_n \quad n = 1, \dots, N \quad (3.2)$$

### 3.3.1. Calificación de la recuperación

En esta sección describimos el procedimiento de evaluación de la recuperación diaria de cara al entrenamiento, especificando qué parámetros conformarán la rúbrica, con qué criterio se calificará cada parámetro y qué peso corresponderá a cada uno en la evaluación de la recuperación.

#### Parámetros considerados para la evaluación

Para determinar qué parámetros son considerados para la calificación de la recuperación, se hace una distinción entre los tipos de días de la semana de estudio según el tipo de actividad llevada a cabo ese día y el día anterior. En otras palabras, distinguimos si es un día de entrenamiento o de descanso y, en el primer caso, de qué tipo fue el día anterior.

El razonamiento seguido, en el que para cada día se hace una distinción entre la recuperación frente a la fatiga acumulada del Día Anterior (DA) y la recuperación durante el día en el que se realiza el entrenamiento, es el siguiente:

- **Días de descanso.** Se considera que la fatiga del día anterior es paliada con el descanso, por lo que no se incluye. Tendremos en cuenta los parámetros de evolución diaria ajenos al entrenamiento. Nos interesa especialmente analizar el nivel de energía con el que el usuario ha acabado el día, para comprobar que la relajación ha sido buena. También consideramos los parámetros medidos por la mañana para analizar los efectos del sueño en la recuperación del organismo.
- **Días de entrenamiento tras día de descanso.** La posible fatiga del día anterior se evalúa con la calificación total del día de descanso posterior. También contaremos con la calidad del sueño y los parámetros basales por la mañana. A partir de ahí, tendremos en cuenta los parámetros que nos permiten analizar la preparación del organismo previa al entrenamiento.
- **Días de entrenamiento tras día de entrenamiento.** Para analizar la fatiga del día anterior, analizaremos la recuperación post-entrenamiento de la sesión del día previo. El razonamiento de cara al entrenamiento del propio día es el mismo que para los días con entrenamiento tras día de descanso.

Así, los parámetros considerados según el tipo de día son los siguientes. En la tabla se distingue con colores la recuperación frente a la fatiga del DA (amarillo) y la del propio día (verde).:

Día de Descanso	Día de Entrenamiento tras Descanso	Día de Entrenamiento tras Entrenamiento
Parámetros	Parámetros	Parámetros
Calidad del Sueño	Recuperación DA	Estrés Post-Entrenamiento DA
HRV Basal	Calidad del Sueño	BB Final DA
BB Inicial	HRV Basal	Calidad del Sueño
Estrés Diario	BB Inicial	HRV Basal
BB Final	Estrés Pre-Entrenamiento	BB Inicial
	BB Pre-Entrenamiento	Estrés Pre-Entrenamiento
		BB Pre-Entrenamiento

Cuadro 3.5: Parámetros para la evaluación de la recuperación por tipo de día

### Criterio de calificación de cada parámetro

El criterio de calificación se basa en una adaptación del sistema que utiliza Garmin para calificar sus parámetros a los valores de referencia obtenidos a partir de la revisión de la literatura de deportes de contacto del Capítulo 2. A modo de recordatorio, se resume este planteamiento:

- La **calidad del sueño** se califica con el propio sistema de calificación de Garmin, que sigue nuestro sistema de cuatro niveles.
- El **HRV** se califica a partir del rango basal del usuario:

$$HRV_B = [HRV_B[0], HRV_B[1]] \quad (3.3)$$

Nuestro HRV (*hrv*) será excelente si está en la mitad superior del umbral y bueno si está en la inferior. Será regular si se sale del umbral por encima o por debajo a una distancia de 10 (desbalanceado para Garmin) y malo en el resto de casos.

- Los límites de calificación de la **BB inicial** ( $bb_i$ ) coinciden con los límites de categorización de Garmin (2.6, solo que el límite de excelencia se ha elevado ya que por la mañana estos valores han de ser más altos (recordamos que el sistema de calificación de Garmin considera la BB de todo el día). Los límites de calificación del **BB pre-entreno** ( $bb_{pre}$ ) y **BB final** ( $bb_f$ ) se han ajustado teniendo en cuenta que el valor diario disminuye a lo largo del día (los límites de calificación disminuyen proporcionalmente con los valores obtenidos).
- Los límites de calificación del **nivel de estrés** ( $e$ ) coinciden con los límites de categorización de Garmin (2.7).

En la siguiente tabla, que recoge el sistema de calificación que seguiremos, se especifica si los valores usados son de las referencias de la literatura [R] o del sistema de calificación de Garmin [G]. Se añadirá una tercera marca, [E], a los valores umbral que hemos estimado a partir de los datos capturados y una cuarta, [RE], para los valores que hemos estimado a partir de valores de referencia.

Parámetro	Calificación	Criterio
Calidad del Sueño	Sistema de calificación de sueño de Garmin [G]	
HRV	Excelente	$hrv \in [\frac{HRV_B[1]+HRV_B[0]}{2}, HRV_B[1]]$ [E]
	Bueno	$hrv \in [HRV_B[0], \frac{HRV_B[1]+HRV_B[0]}{2})$ [E]
	Regular	$hrv \in [HRV_B[0] - 10, HRV_B[0]] \cup (HRV_B[1], HRV_B[1] + 10)$ [G]
	Malo	$hrv < HRV_B[0] - 10$ ó $hrv > HRV_B[1] + 10$ [G]
Body Battery Inicial	Excelente	$bb_i >= 85$ [E]
	Bueno	$bb_i >= 50$ [G]
	Regular	$bb_i >= 25$ [G]
	Malo	$bb_i < 25$ [G]
Body Battery Pre-Entreno	Excelente	$bb_{pre} >= 50$ [E]
	Bueno	$bb_{pre} >= 35$ [E]
	Regular	$bb_{pre} >= 20$ [E]
	Malo	$bb_{pre} < 20$ [E]
Body Battery Final	Excelente	$bb_f >= 25$ [E]
	Bueno	$bb_f >= 15$ [E]
	Regular	$bb_f >= 10$ [E]
	Malo	$bb_f < 10$ [E]
Nivel de Estrés	Excelente	$e <= 25$ [G]
	Bueno	$e <= 50$ [G]
	Regular	$e <= 75$ [G]
	Malo	$e > 75$ [G]

Cuadro 3.6: Criterio de calificación de los parámetros de evaluación de la recuperación

### Peso de cada parámetro

La asignación de pesos se plantea haciendo una interpretación de la literatura estudiada.

A la recuperación respecto al día anterior, considerada en los días de entrenamiento, se le da un peso de 2, asignado de forma íntegra para la calificación global de la recuperación del día de descanso para los días de entrenamiento tras descanso, o dividida entre el estrés post-entrenamiento y la batería corporal del día anterior en el caso de días de entrenamiento tras entrenamiento.

La calidad del sueño se ha asumido de gran importancia según la evidencia científica, por lo que se le da un peso de 2, pero su reflejo en la energía por la mañana hace que a los parámetros monitorizados al despertar se les dé una ponderación de 3. Esta se reparte en 2 para el HRV basal (con mayor peso científico) y 1 para el BB inicial.

A los parámetros de estrés diario (para días de descanso) y pre-entrenamiento (para días de entrenamiento) se les asigna un peso 1, debido a la menor fiabilidad de los datos capturados, como se comentará en el Capítulo 5.

Finalmente, se le ha dado gran importancia a la energía con la que el usuario llega al entrenamiento, medida con el Body Battery Pre-entrenamiento, con una ponderación de 2 puntos.

En la siguiente tabla se recoge esta ponderación:

Día de Descanso		Día de Entrenamiento tras Descanso		Día de Entrenamiento tras Entrenamiento	
Parámetro	Peso	Parámetro	Peso	Parámetro	Peso
Calidad del Sueño	2	Recuperación DA	2	Estrés Post-Entrenamiento DA	1
HRV Basal	2	Calidad del Sueño	2	BB Final DA	1
BB Inicial	1	HRV Basal	2	Calidad del Sueño	2
Estrés Diario	1	BB Inicial	1	HRV Basal	2
BB Final	2	BB Pre-Entrenamiento	1	BB Inicial	1
		BB Pre-Entrenamiento	2	Estrés Pre-Entrenamiento	1
				BB Pre-Entrenamiento	2

Cuadro 3.7: Tabla de pesos de cada parámetro por tipo de día.

### Calificación global de la recuperación

Finalmente, calificamos la recuperación con el sistema de cuatro niveles, tal que:

Calificación	Puntuación
Excelente	$\geq 3.5$
Buena	$\geq 2.5$
Regular	$\geq 1.5$
Mala	$< 1.5$

Cuadro 3.8: Criterio de calificación global de la recuperación

### 3.3.2. Calificación del rendimiento deportivo

Este apartado recoge, de la misma manera, los parámetros a tener en cuenta para la evaluación del rendimiento deportivo, así como el criterio seguido para su calificación y los pesos de cada uno en el cálculo de la puntuación global.

#### Parámetros considerados para la evaluación

El planteamiento propuesto para la selección de parámetros que confeccionan la rúbrica de evaluación del rendimiento deportivo, siguiendo la literatura de deportes de contacto revisada en el Capítulo 2, es el siguiente:

- Tendremos en cuenta el  $\text{VO}_2$  máximo de cada usuario a la hora de calificar el parámetro, siendo menos estrictos con los sujetos de menor capacidad cardiovascular. Este parámetro, por tanto, no se incluye directamente en la rúbrica, sino que su valor modificará los límites de puntuación de la calificación global.
- Valoraremos el rendimiento con los parámetros de frecuencia cardíaca, comprobando si la FC media está dentro de las zonas de FC óptima y si no se supera el valor de referencia para los asaltos de actividad.
- Valoraremos también la recuperación entre asaltos, analizando que no se supera el valor de FC referencia para asaltos de recuperación. Además, se comprobará si el usuario es capaz de recuperar su frecuencia respiratoria previa al entrenamiento.

Parámetros
VO <sub>2</sub> Máximo
Frecuencia cardíaca media
Frecuencia cardíaca en asaltos de actividad
Frecuencia cardíaca en asaltos de recuperación
Balance de frecuencia respiratoria pre/post entrenamiento

Cuadro 3.9: Parámetros para la evaluación del rendimiento deportivo

### Criterio de calificación de cada parámetro

El criterio seguido para la calificación de los parámetros de manera resumida, recordando los conceptos revisados de la literatura del Capítulo 2, es el siguiente:

- El **VO<sub>2</sub> máximo** (*vo<sub>2max</sub>*) se califica con el sistema tabulado por Garmin (tabla 2.5), adaptado para que el excelente se obtenga si se alcanza el valor de referencia de los estudios de la literatura.
- Para valorar la **FC media** durante el entrenamiento se evalúa comprobando si está dentro de la ZFCO, referenciada en la tabla por sus límites como:

$$ZFCO = [ZFCO_{75}, ZFCO_{80}] \quad (3.4)$$

La calificación es excelente si el valor registrado (*fc<sub>media</sub>*) está dentro de la ZFCO con un margen de 5 bpm desde cada límite y buena si está dentro de la ZFCO pero fuera de ese margen. Se obtiene una calificación regular si se encuentra fuera de la zona óptima a una diferencia menor de 5 bpm (por encima o por debajo) y una calificación mala si la diferencia es mayor.

- La **frecuencia cardíaca** media durante los **asaltos de actividad y recuperación** se valora a partir del valor de referencia (véase la sección 2.2.3), siendo excelente si la FC registrada (*fc<sub>asaltos</sub>*) está muy por debajo, bueno si está sobre ese valor y regular o malo dependiendo de cuán por encima se encuentre de dicha referencia.
- Finalmente, el **balance de FR pre/post entrenamiento** se califica según si el valor obtenido ( $\Delta fr$ ) se aleja a unas distancias establecidas según los valores experimentales obtenidos, asociando dichos valores a las sensaciones del usuario.

Siguiendo el mismo formato que para la recuperación:

Parámetro	Calificación	Criterio
VO <sub>2</sub> Máx	Excelente	$vo_2max \geq 54.5$ [R]
	Bueno	$vo_2max \geq 45.4$ [G]
	Regular	$vo_2max \geq 41.7$ [G]
	Malo	$vo_2max < 41.7$ [G]
FC media	Excelente	$fc_{med} \in [ZFO_{70} + 5, ZFO_{85} - 5]$ [E]
	Buena	$fc_{med} \in [ZFO_{70}, ZFO_{70} + 5] \cup (ZFO_{85} - 5, ZFO_{85}]$ [E]
	Regular	$fc_{med} \in [ZFO_{70} - 5, ZFO_{70}] \cup (ZFO_{85}, ZFO_{85} + 5]$ [E]
	Malo	$fc_{med} < ZFO_{70} - 5$ ó $fc_{med} > ZFO_{85} + 5$ [E]
FC por Asaltos	Excelente	$fc_{casaltos} < V_{Ref} - 5$ [RE]
	Bueno	$fc_{casaltos} \leq V_{Ref} + 2$ [RE]
	Regular	$fc_{casaltos} \leq V_{Ref} + 10$ [RE]
	Malo	$fc_{casaltos} > V_{Ref} + 10$ [RE]
Balance FR pre/post entrenamiento	Excelente	$\Delta fr \leq 0.5$ [E]
	Bueno	$\Delta fr \leq 1$ [E]
	Regular	$\Delta fr \leq 1.5$ [E]
	Malo	$\Delta fr > 1.5$ [E]

Cuadro 3.10: Criterio de calificación de los parámetros de evaluación del rendimiento

### Peso de cada parámetro

A continuación se definen los pesos que le daremos a cada parámetro para la evaluación del rendimiento.

Tras la revisión de la literatura realizada en el Capítulo 2 de este informe, se ha concluido que entrenar dentro del rango de pulsaciones que conforma la ZFCO, calculado a partir de la HRR según el método de Karvonen, es un buen indicador del rendimiento del entrenamiento, además que su estudio engloba a deportistas casuales (posibles usuarios) y no solo a boxeadores. Por tanto, nos centraremos en este parámetro para la evaluación de la frecuencia cardíaca y el peso la calificación de la FC media durante el entrenamiento será el mayor.

Por su parte, a las frecuencias cardíacas umbral para los asaltos de actividad y descanso se les asignará un peso de 1.

Finalmente, la gran importancia del control de la respiración para disminuir el efecto de la fatiga cardiovascular en deportes de contacto, comentada en la revisión de la literatura, hace que se haya decidido dar al balance de respiración pre/post entrenamiento un gran peso.

Los pesos comentados se recogen en la siguiente tabla:

Parámetro	Peso
Frecuencia cardíaca media	3
Frecuencia cardíaca en asaltos de actividad	1
Frecuencia cardíaca en asaltos de recuperación	1
Balance de frecuencia respiratoria pre/post entrenamiento	2

Cuadro 3.11: Parámetros para la evaluación del rendimiento deportivo

### Calificación global del rendimiento

En este caso, los límites de puntuación para la calificación global del rendimiento están influenciados por el valor del VO<sub>2</sub> máximo del usuario, de tal manera que se será más estricto cuanto mejor sea el VO<sub>2</sub> máximo:

Calificación VO <sub>2</sub> Máx.	Calificación	Puntuación
Excelente	Excelente	$\geq 3.5$
	Bueno	$\geq 2.5$
	Regular	$\geq 1.5$
	Malo	$< 1.5$
Bueno	Excelente	$\geq 3.4$
	Bueno	$\geq 2.4$
	Regular	$\geq 1.4$
	Malo	$< 1.4$
Regular	Excelente	$\geq 3.3$
	Bueno	$\geq 2.3$
	Regular	$\geq 1.3$
	Malo	$< 1.3$
Malo	Excelente	$\geq 3.2$
	Bueno	$\geq 2.2$
	Regular	$\geq 1.2$
	Malo	$< 1.2$

Cuadro 3.12: Criterio de calificación global del rendimiento en función del VO<sub>2</sub> máximo

### 3.3.3. Evaluación de la eficiencia del algoritmo

La limitación de tiempo y de recursos (dispositivos wearables) ha hecho que no se pueda evaluar la eficiencia del algoritmo mediante un piloto con un número amplio de sujetos de estudio (como se expone en la sección de análisis de cumplimiento de los objetivos, 6.1).

Por tanto, a falta de la ampliación del número de sujetos en investigaciones futuras (6.2), se evaluará cualitativamente el funcionamiento del algoritmo comprobando si los resultados obtenidos del análisis coinciden con las sensaciones y las observaciones de los usuarios de cara al estado previo (recuperación) y el rendimiento del entrenamiento. Esto nos dará un posible contexto a los resultados obtenidos.

Estos aspectos serán registrados antes y después de cada sesión de entrenamiento con la siguiente rúbrica:

Aspecto	Sensaciones/Observaciones
Fatiga acumulada	
Energía previa al entrenamiento	
Desempeño en el entrenamiento	
Intensidad del entrenamiento	
Recuperación y respiración entre asaltos	

Cuadro 3.13: Aspectos sobre los que se registran sensaciones y observaciones de los usuarios

Donde las sensaciones de **fatiga acumulada** y **energía previa al entrenamiento** nos permitirán dar contexto a la calidad de la recuperación de cara al entrenamiento; el **desempeño** y la **intensidad del entrenamiento** nos dará información extra sobre los resultados asociados a los parámetros de frecuencia cardíaca (si un valor es anormal porque se ha entrenado a una intensidad menor de la debida por 'falta de ganas'); y la **recuperación y respiración entre asaltos** nos podría permitir justificar los resultados de balance respiratorio.

### 3.4. Encuesta SUS para la valoración de la usabilidad

Para valorar la usabilidad del sistema de monitorización propuesto, se ha realizado una encuesta SUS sobre la red personal a cada uno de los usuarios, orientada al cumplimiento de los principios de los dispositivos wearables descritos en la sección 2.1.1.

La encuesta SUS (System Usability Scale) o escala de usabilidad de sistema, es un método de análisis de la facilidad de uso de un sistema, producto o servicio diseñado en 1986 por Jonh Brooke y, a día de hoy, sigue siendo uno de los estándares de valoración de usabilidad en el mercado.

Consiste en un cuestionario de 10 preguntas estándar (válidas para todo tipo de productos) que se responden con un sistema de puntuación del 1 al 5, donde 1 significa 'totalmente en desacuerdo' y el 5 'totalmente de acuerdo', a modo de escala Likert.

Las cuestiones son las expuestas a continuación. Cada pregunta se matizó para ser orientada a las características fundamentales de la tecnología wearable, evaluando su cumplimiento desde la perspectiva del usuario. Estos matices se recogen en la tabla 3.14:

1. Creo que me gustaría usar este sistema con frecuencia.
2. Encontré el sistema innecesariamente complejo.
3. Pensé que el sistema era fácil de usar.
4. Creo que necesitaría el apoyo de un técnico para poder usar este sistema.
5. Encontré que las diversas funciones del sistema estaban bien integradas.
6. Pensé que había demasiada inconsistencia en este sistema.
7. Imagino que la mayoría de las personas aprenderían a usar este sistema muy rápidamente.
8. Encontré el sistema muy incómodo de usar.
9. Me sentí muy seguro(a) usando el sistema.
10. Necesité aprender muchas cosas antes de poder empezar a usar este sistema.

Cuestión	Característica Fundamental de Dispositivo Wearable
1, 8	Cumple con los requisitos de <b>monitorización, portabilidad y autonomía</b> propuestos.
2, 3, 4, 7, 10	¿Su uso es accesible y no intrusivo? Se valora la <b>portabilidad y la autonomía</b> de uso.
5, 6, 9	¿Sentiste que el sistema de <b>monitorización y conectividad</b> es fiable?

Cuadro 3.14: Orientación de la encuesta SUS a la tecnología wearable

Tras la encuesta, se calcula una puntuación numérica del 0 al 100 y a la misma se le asigna una calificación, que vemos en la tabla 3.15. La operación para calcular la operación es la siguiente:

1. A la respuesta de las preguntas impares se les resta 1 punto.
2. Se resta a 5 la respuesta de las preguntas pares.
3. El resultado se suma y se multiplica por 2,5.

Puntuación SUS	Calificación SUS
78,9 – 100	A
72,6 – 78,8	B
62,7 – 72,5	C
51,7 – 62,6	D
0 – 51,6	F

Cuadro 3.15: Calificación SUS según la puntuación SUS.

Finalmente, se establece un nivel de aceptabilidad según la puntuación obtenida.

Puntuación SUS	Nivel de Aceptabilidad
70 – 100	Aceptable
50 – 69	Marginal
0 – 49	No aceptable

Cuadro 3.16: Nivel de aceptabilidad según puntuación SUS



# 4. Desarrollo Experimental y Técnico

En este capítulo se expone el proceso de desarrollo seguido para poner en práctica los principios teóricos que fundamentan este Proyecto.

Se describe tanto el planteamiento del sistema de monitorización con la red personal de dispositivos wearables desplegada para la captación de los datos de estudio, como la implementación técnica del sistema de análisis de datos y el algoritmo propuesto para la calificación del rendimiento y la recuperación deportiva.

## 4.1. Desarrollo Experimental

Esta sección esta dedicada a describir el procedimiento diseñado para la monitorización diaria de los usuarios para la captación de los datos de estudio.

### 4.1.1. Procedimiento a seguir por el usuario

En esta sección se describirá, a modo de guía de uso, el procedimiento diario a seguir por el usuario para la monitorización de su rendimiento deportivo y su recuperación de usuario con la red personal de dispositivos wearables diseñada.

Este procedimiento se fundamenta en las características de los dispositivos de nuestra red personal, referentes tanto a las limitaciones físicas marcadas por el entrenamiento boxístico expuestas en el capítulo 2.1.4, como a la manera en la que los sensores que incorporan miden los parámetros de interés, listados en la sección 2.2.3.

Así, se ha diseñado un procedimiento de monitorización de una semana de duración, diferenciado en tres tipos de días. Los días en los que se entrena boxeo están compuestos por tres fases diarias, divididas por el periodo de entrenamiento, ya que los parámetros medidos diferirán según el usuario este entrenando o no. Además, añadimos una fase inicial preparatoria previa a la semana de medición.

- **Días de Preparación.** Correspondiente a las mediciones que deberán realizarse de manera previa a la semana de entrenamiento.
- **Días de Entrenamiento.** Los cuatro días semanales en los que hay entrenamiento de boxeo se realiza un seguimiento dividido en tres fases, divididas precisamente por dicha sesión.
  - **Fase Pre-entrenamiento.** Primera fase diaria de los días de entrenamiento de boxeo. Comprendida desde el inicio del día hasta antes del entrenamiento.
  - **Fase de Entrenamiento.** Segunda fase diaria de los días de boxeo. Corresponde a la monitorización de la sesión de entrenamiento.

- **Fase Post-entrenamiento.** Fase final diaria de los días de entrenamiento de boxeo, comprendida desde la finalización del mismo hasta que el usuario se acuesta.
- **Días de Descanso.** Referente a las acciones que el usuario deberá realizar los dos días de descanso de la semana.

En cada uno de estos días y sus respectivas fases, se incluye una tabla a modo de itinerario en el que indican las mediciones que el usuario tendrá que realizar de manera manual mediante acciones con su reloj inteligente (y cuándo y cómo lo hará) y las que, por el contrario, se obtendrán de manera automática.

Se colorearán las celdas correspondientes a las acciones manuales (actividades y pruebas, 3.2.2), siguiendo el código de colores mostrado a continuación:

Prueba	Color
Prueba de Carrera	Amarillo
Resumen de Salud	Verde
Barra (Entrenamiento)	Lila

Cuadro 4.1: Caption

### Días de Preparación.

La primera parte del proceso consiste en registrar los parámetros personales de referencia que usaremos como valores umbral para el estudio. Este procedimiento es previo a la semana de entrenamiento.

Una semana antes de comenzar el estudio, se registrará con el reloj los valores de **frecuencia cardíaca basal**, para lo que simplemente lo llevará puesto durante varios días ya que Garmin separa este valor, y el **HRV en reposo**, realizando tres de **Resumen de Salud** (véase la sección 3.2.2) a lo largo del día durante tres días, lo que nos permitirá determinar un umbral del valor típico.

Por otro lado, el día previo a la semana de entrenamientos se centra exclusivamente en la medición del VO<sub>2</sub> Máximo del usuario con antelación al estudio. Aunque es ajeno a la semana del entrenamiento, el usuario comenzará a llevar Este parámetro, como se ha comentado, es calculado por Garmin con sus métricas propias a partir de una **Prueba de Carrera** con el reloj inteligente, recomendablemente (por precisión) de al menos 15 minutos, por lo que el usuario deberá realizar esta prueba al aire libre el día de antes de iniciar la semana de entrenamientos de boxeo.

Además, aprovecharemos para comenzar a medir el periodo de sueño de la noche previa al día 1 de monitorización, procedimiento correspondiente a la Fase Pos-entrenamiento. Para esta medición, el usuario dormirá con su smartwatch. Durante este periodo, comenzará a medirse el balance calórico del día siguiente.

Así, el itinerario de la fase inicial del Proyecto es el siguiente:

Momento del día (hora)	Medición manual [Acción (parámetro)]	Medición autónoma
NE (NE)	Prueba de Carrera (VO2 Máx)	
Periodo de sueño (NE)		Calidad del sueño, balance calórico

Cuadro 4.2: Plan de monitorización de la Fase de Preparación

**Día de Entrenamiento**

- **Fase Pre-entrenamiento.** Comprendida desde que el usuario se despierta hasta que comienza su sesión de entrenamiento.

Durante esta primera fase del día, el usuario deberá llevar equipado su smartwatch para obtener datos de manera continua y manual. El proceso se describe a continuación y se muestra en la tabla x:

- Medición autónoma. La medición autónoma del Venu 2 permite hacer un seguimiento continuo de parámetros que nos interesa cuantificar de manera diaria, como el **balance calórico**, o cuya variación a lo largo del día es de interés para ver qué efectos tiene sobre el entrenamiento, como el **estrés** y la **batería corporal**.
- Medición manual. En la fase previa al entrenamiento, nos interesa el impacto que la recuperación al ejercicio del día anterior va a afectar en el rendimiento deportivo del usuario. Este aspecto, como se ha comentado en este trabajo, se refleja en el valor de la variabilidad de la frecuencia cardíaca o **HRV**.

Para medir el HRV, se realizará, de manera previa al entrenamiento, dos pruebas de **Resumen de Salud** en nuestro reloj Garmin. La primera será realizada en cuanto el usuario se despierte (2.2.3) y la segunda de manera sistemática a medio día (14:00h), para analizar su evolución antes del momento de actividad.

Momento del día (hora)	Medición manual [Acción (parámetro)]	Medición autónoma
Al levantarse (NE)	Resumen de Salud (HRV)	
Medio día (14:00h)	Resumen de Salud (HRV)	Balance calórico, estrés, batería corporal
Fase de Entrenamiento		

Cuadro 4.3: Plan de monitorización de la Fase Pre-entrenamiento

- **Fase de Entrenamiento.** Referente al periodo del día dedicado expresamente a la sesión de entrenamiento de boxeo.

Su duración es de 90 minutos y sigue la estructura interválica HIIT previamente descrita. Esta estructura dividida en intervalos de actividad y de descanso, como se ha comentado, limita el uso del smartwatch únicamente a los intervalos de recuperación, en los que el reloj sí que registrará datos.

Por ello, el usuario deberá equiparse ambos dispositivos, tanto el reloj como la banda pectoral, durante la totalidad de la sesión. La puesta a punto de la red personal para el entrenamiento se recoge en la sección '4.1.2'.

Esta vez, la toma de datos será manual, mediante el uso de aplicaciones y pruebas de Garmin. Concretamente (y una vez se ha sincronizado el sensor de frecuencia cardiaca con el smartwatch), el usuario deberá hacer 3 acciones con su reloj inteligente:

1. En primer lugar, justo antes de comenzar el entrenamiento registramos el **HRV** del usuario mediante un Resumen de Salud, para ver como varía en su transcurso.
2. Para monitorizar los datos del entrenamiento, el usuario deberá iniciar la actividad personalizada 'Barra' (véase la sección 3.2.2), que medirá parámetros como el **estrés**, la **frecuencia respiratoria** y las **calorías** consumidas, además de la **frecuencia cardiaca**, que tomará sus valores con la tecnología ECG de la Garmin HRM-Pro.  
Una vez acabado el entrenamiento, el usuario deberá detener la actividad y guardar los dato captados.
3. Finalmente, volveremos a capturar el valor del HRV post-entrenamiento para poder compararlo al obtenido al inicial. Para ello el usuario volverá a tomar una prueba de Resumen de Salud.

La siguiente tabla recoge el itinerario de esta fase:

Momento del día (hora)	Medición manual [Acción (parámetro)]
Antes del entrenamiento (NE)	Resumen de Salud (HRV, estrés y FR)
Entrenamiento 90”(NE)	Barra (Frecuencia respiratoria, frecuencia cardiaca, calorías)
Al finalizar el entrenamiento (NE)	Resumen de Salud (HRV, estrés y FR)

Cuadro 4.4: Plan de monitorización de la Fase de Entrenamiento

- **Fase Post-entrenamiento.** La fase final de los días de medición la semana de boxeo comprenden desde la finalización del entrenamiento hasta que el usuario se acuesta. En esta fase se llevará equipado de manera continua el reloj inteligente.

Seguiremos la estructura de monitorización autónoma y manual de la Fase Pre-entrenamiento:

- Medición autónoma. Hasta la hora de dormir, nos interesa seguir contando las calorías quemadas para hacer un posterior recuento diario. Seguiremos monitorizando el estrés, que puede influir en la calidad del sueño, y la batería corporal.  
El usuario dormirá con el reloj puesto para monitorizar su sueño de manera autónoma, lo cual nos podrá dar información de valor de la recuperación del entrenamiento de ese día de cara al día siguiente. Durante este periodo también se hace recuento calórico.
- Medición manual. La medición manual de esta fase corresponde a un último Resumen de Salud para obtener el valor del HRV antes de acostarnos. Así, tendremos un conjunto de valores que describen la evolución de este parámetro a lo largo de un día de entrenamiento.

Finalmente, la siguiente tabla recoge el itinerario de la última fase de la monitorización diaria:

Momento del día (hora)	Medición manual [Acción (parámetro)]	Medición autónoma
Tras el entrenamiento (NE)		Balance calórico, estrés, batería corporal.
Antes de acostarse (NE)	Resumen de Salud (HRV)	
Periodo de sueño (NE)		Calidad del Sueño, balance calórico.

Cuadro 4.5: Plan de monitorización de la Fase Post-entrenamiento

### Día de Descanso

Como se expuso en la sección 2.2.2, la rutina en la que se basa este estudio cuenta con dos días de descanso.

Durante este día, el usuario únicamente realizará los tres resúmenes de salud rutinarios (nada más despertarse, a medio día y al acostarse), ya que nos interesa la evolución del HRV durante el día, pero sobretodo el balance entre valor inicial y el obtenido al acabar la jornada, ya que podremos compararlo con el registrado los días de entrenamiento para analizar el impacto del mismo.

Los parámetros medidos de manera autónoma son los mismo que los obtenidos los otros días en las fases ajenas al entrenamiento.

Así, el itinerario de estos días es el reflejado en la siguiente tabla:

Momento del día (hora)	Medición manual [Acción (parámetro)]	Medición autónoma
Al levantarse (NE)	Resumen de Salud (HRV)	
Medio día (14:00h)	Resumen de Salud (HRV)	Balance calórico, estrés, batería corporal
Medio día (14:00h)	Resumen de Salud (HRV)	

Cuadro 4.6: Plan de monitorización de los días de descanso

#### 4.1.2. Configuración y puesta a punto de los dispositivos

En esta sección se describe los ajustes necesarios para la puesta a punto del modo de operación de los dispositivos durante la fase de captación de datos, desde la configuración software para ajustar el modo de medición hasta su disposición física durante los entrenamientos.

- **Configuraciones software.**

- **Activación del modo de Grabación Por Segundo.** Garmin cuenta con una funcionalidad que permite el guardado de datos cada segundo para una monitorización continua del usuario, frente al modo por defecto 'Smart Recording' o Grabación Inteligente, enfocado en el ahorro de batería del reloj.

Para activar esta funcionalidad, desde la cuenta Garmin Connect asociada a nuestro reloj accederemos a los ajustes generales del dispositivo y activaremos Grabación Por Segundo en el apartado 'Grabación de Datos'.

Recording Activities

Data Recording  Smart Recording (Recommended)  Every Second Recording

Figura 4.1: Configuración del modo de grabación del Venu 2 desde Garmin Connect

- **Sincronización del sensor de frecuencia cardiaca.** Para que los datos de FC obtenidos sean los capturados por el sensor de frecuencia cardíaca, cada vez que el usuario vaya a iniciar su entrenamiento deberá sincronizar ambos dispositivos.  
Para esto, se sigue el proceso descrito en la sección 3.1.4.
- **Configuraciones físicas.** En este apartado describimos la colocación de los dispositivos wearables durante el entrenamiento, para protegerlos frente a daños en la medida de lo posible.
  - **Garmin HRM-Pro Plus.** El sensor de frecuencia cardiaca irá colocado sobre la piel, debajo de la ropa, de tal manera que el propio sensor quedé justo debajo del esternón. Lo ajustaremos bien con su sistema de hebillas, de tal manera que quede inmóvil durante todo el entrenamiento.  
Se activará de forma automática cuando detecte el contacto con la piel.
  - **Garmin Venu 2.** Nuestro reloj inteligente está equipado en una zona más sensible, con mayor disposición a golpes, tanto recibidos como lanzados por el propio usuario. Por ello, intentamos protegerlo en mayor medida. Además, procuraremos que la precisión de sus sensores se vea afectada lo menos posible al movimiento continuo.  
Se colocará muy ajustado, un punto más de la medida correspondiente a la medida de la muñeca del usuario, para intentar que los sensores puedan tomar medidas durante el entrenamiento (recordemos que las medidas de estrés y frecuencia cardiaca son sensibles al movimiento).  
La protección del reloj, por otro lado, comienza con la incorporación de un protector de pantalla de silicona. Continuaremos usando el vendaje de boxeo sobre el reloj, desde la pantalla a la correa, para ajustarlo y protegerlo. Por último, al colocarnos los guantes de boxeo comprobaremos que el reloj está bien sujetado y que queda debajo de su cierre de velcro, que actuará también de protector.



Figura 4.2: Protector de pantalla de silicona y vendaje de boxeo.

En la siguiente figura podemos comprobar cómo quedarían equipados ambos wearables para una sesión de entrenamiento (mostramos el reloj sin el guante de boxeo puesto):



Figura 4.3: Disposición de los dispositivos wearables para el entrenamiento.



Figura 4.4: Disposición de los dispositivos wearables para el entrenamiento.

#### 4.1.3. Usuarios voluntarios del estudio

En la realización del Proyecto, participaron dos sujetos usuarios de la red personal de dispositivos wearables para la captación de datos de análisis.

Antes de comenzar el estudio, firmaron un consentimiento informado en el que se indica que sus datos serán utilizados para la realización del estudio de investigación, por lo que se comprometen a realizar las pruebas específicas de captación, consistentes en una primera etapa de determinación de parámetros basales y una segunda fase para el seguimiento de los entrenamientos y la recuperación frente a ellos durante una semana.

Estos datos se tratarán de forma anónima para cumplir con la Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPDGDD). De esta forma, los usuarios estarán identificados únicamente por un número de usuario.

#### Características de los usuarios

En este apartado se describen las características físicas de los sujetos que serán monitorizados durante este procedimiento, a partir de los cuales obtendremos los datos para su posterior análisis.

Las características físicas a tener en cuenta son las siguientes:

- Edad, género, altura y peso. Incluidas como variables condicionales en numerosos estudios de la literatura de los deportes de contacto.

Estos datos serán medidos en farmacia.

- Nivel de boxeo y contexto deportivo. Recogemos el nivel de entrenamiento de los sujetos en la disciplina que vamos a monitorizar, así como su contexto deportivo para entender su situación física y cardiovascular en el momento en el que inician el estudio (si realizan otro deporte, si vienen de un periodo sedentario, etc.). Estos aspectos podrán reflejarse en los datos obtenidos y nos servirán de contexto para el análisis.

## Usuario 1

El Usuario 1 es un varón de 19 años estudiante de magisterio, que entrena 2 horas diarias en el gimnasio y cuida su alimentación, pero nunca ha practicado boxeo. Aunque sea deportista habitual, es asmático por lo que su capacidad cardiorrespiratoria no es muy alta.

La siguiente tabla lista las características del Usuario 1.

Parámetro	Valor
Género	Masculino
Edad	19 años
Peso	83,50 kg
Altura	1,94 m
Nivel de Boxeo	Principiante
Contexto Deportivo	Entrena en gimnasio

Cuadro 4.7: Características físicas del usuario 1

## Usuario 2

El Usuario 2 es un varón de 21 años estudiante al mismo tiempo de un grado de ingeniería y de un máster de Big Data. Cuida mucho su alimentación y practica boxeo y levantamiento de pesas en el gimnasio, pero su altas horas de estudio hace que no siga un entrenamiento muy regular.

Las características del segundo usuario se muestran en la siguiente tabla.

Parámetro	Valor
Género	Masculino
Edad	21
Peso	86,00 kg
Altura	1,93 m
Nivel de Boxeo	Amateur
Contexto Deportivo	5 meses de boxeo

Cuadro 4.8: Características físicas del usuario 2

## 4.2. Desarrollo Técnico

En esta sección se recoge el proceso de implementación en código de los sistemas planteados para la consecución de los objetivos del Proyecto.

*Aclaración: El formato de los cuadros en los que mostramos el código en esta sección no permite acentos ni caracteres especiales como la ‘ñ’, por lo que los evitamos su uso. Se escriben las palabras sin tilde y se sustituye ‘ñ’ por ‘n’.*

#### 4.2.1. Desarrollo fallido de una aplicación Garmin Connect IQ

El primer planteamiento de los objetivos del Proyecto concluyó con la posibilidad de desarrollar una aplicación ejecutable en el propio smartwatch, bajo la plataforma de desarrollo Garmin Connect IQ que se encargara de almacenar las métricas relevantes específicamente en las pruebas de entrenamiento.

El entorno de desarrollo elegido fue Visual Studio Code, con la extensión de Monkey C, que se utiliza por defecto en la infraestructura Connect IQ de Garmin para el desarrollo de aplicaciones para sus relojes.

La aplicación ideada consistía en un entrenamiento interválico de alta intensidad (HIIT), que permitiera configurar la duración de los intervalos de actividad y de recuperación, así como la duración total de la sesión de entrenamiento, devolviendo los resultados diferenciados entre tipo de intervalo. Estos datos se irían almacenando para luego poder evaluarlos.

A continuación muestro el boceto inicial de la aplicación que fue presentado al tutor:



Figura 4.5: Boceto inicial de la aplicación Connect IQ fallida

El proceso de desarrollo duró 4 semanas incluyendo el tiempo requerido para la formación y familiarización con los métodos del lenguaje. Ahora bien, hubo varias razones para no poder concluir el proyecto:

- la documentación del entorno de desarrollo está obsoleta en parte y resulta difícil revisar sus métodos debido a la reciente (y no respaldada) actualización del SDK.
- El cambio del entorno de desarrollo utilizado de Eclipse a VS Code hace complicado en muchos casos el código desarrollado por terceros de aplicaciones y tutoriales.
- las aplicaciones de terceros no tienen compatibilidad con sensores externos y
- Los parámetros accesibles por sus APIs de uso libre no cumplían totalmente con la lista de parámetros requerida.

Dado que el ecosistema de Garmin incluye también un conjunto de aplicaciones base correspondiente a multitud de deportes con la posibilidad extra de que las métricas se puedan almacenar en la nube del propio Garmin, al que a través de un acuerdo podemos tener acceso, se decidió cambiar la idea inicial de crear una aplicación que se encargara de la supervisión del

entrenamiento por seleccionar las aplicaciones instaladas por defecto por el propio Garmin y configurarías para obtener la información más relevante de las métricas

#### 4.2.2. Estructura de la implementación del sistema de análisis de la recuperación y del rendimiento deportivo

En esta sección se describe el proceso completo de implementación en Python del sistema de análisis de la recuperación y el rendimiento deportivo propuesto, que abarca desde los datos brutos obtenidos en la fase de captación, hasta la visualización de los resultados tras su análisis y la aplicación del algoritmo planteado (3.3).

Podemos distinguir tres etapas principales en el proceso:

- **Preparación de los datos para el análisis.** Esta etapa consiste en el procedimiento de limpieza y ordenación de los datos capturados para su correspondiente análisis.
- **Análisis segmentado de los datos.** En esta etapa se realiza un análisis segmentado de los datos procesados en la etapa anterior para proporcionar información de cada métrica que puede ser interpretado por especialistas del sector. A partir de los análisis segmentados se podrá aplicar posteriormente los algoritmos de calificación.
- **Aplicación del algoritmo de calificación de la recuperación y el rendimiento** La última etapa consiste en la calificación del rendimiento y la recuperación de los usuarios mediante la implementación del algoritmo propuesto. Esta etapa finaliza con la visualización de los datos.

En la siguiente figura 4.6 se muestra el diagrama de flujo que describe cada una de las etapas desarrolladas para la implementación del sistema de análisis.

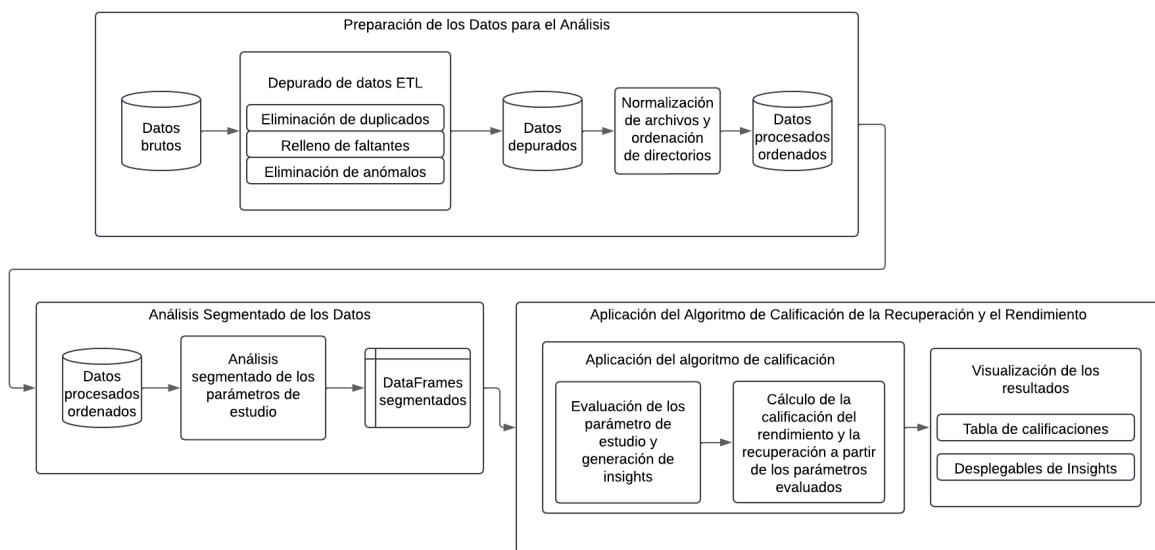


Figura 4.6: Diagrama de flujo de la implementación del sistema de monitorización diseñado

#### 4.2.3. Preparación de los datos para el análisis

En esta sección se muestra el proceso seguido para el depurado inicial de los datos de análisis. Seguiremos un procedimiento **ETL** (Extract, Transform, Load), por lo que se creará una nueva

fuente de datos, es decir, a partir de los archivos JSON originales se generará un nuevo Data Set que posteriormente será cargado y procesado en otro programa.

**La implementación completa del proceso de limpieza de los datos en código Python, así como su explicación detallada, se adjunta en el apéndice A.**

Los datos que se miden de manera continua a lo largo del tiempo, como la frecuencia cardíaca, la frecuencia respiratoria, el estrés o la batería corporal, pueden presentar muestras duplicadas para la misma marca temporal o muestras faltantes en la serie de marcas temporales. El código que se recoge en esta sección propone una solución a esos posibles problemas.

Por otro lado, pueden aparecer datos no significativos por fallos de los sensores, cuyos valores interfieren en la claridad del análisis. Es el caso de los valores de frecuencia cardíaca demasiado altos o bajos, y de los datos de estrés durante el entrenamiento, como se expuso en la sección 2.2.3 de este informe.

El depurado consiste, por tanto, en el tratamiento los tres problemas típicos del depurado del datos, eliminación de duplicados, gestión (en este caso relleno) de datos faltantes y la detección y corrección de valores anómalos o outliers (diferenciados en nuestros dos casos específicos):

- **Eliminación de datos duplicados.** Se localizarán y eliminarán valores duplicados para una misma marca temporal. Para ello se emplean las estructuras *set* de Python, que solo almacenan valores únicos [62].
- **Relleno de datos faltantes.** Se detectarán valores faltantes en la serie temporal y se llenarán de acuerdo a un criterio. En concreto, se utiliza relleno con el valor anterior conocido.
- **Descarte de datos anómalos de FC.** Los valores de frecuencia cardíaca superiores a 220 bpm o inferiores a 40 bpm (rango estimado para deportistas) serán considerados anómalos y se descartarán.
- **Eliminación de datos de estrés captados durante el periodo de entrenamiento.** Se establecerá manualmente el periodo de actividad de los días en los que hay entrenamiento. Durante ese periodo se eliminarán los posibles datos de estrés.

Tras la depuración de los datos brutos, estos se reorganizaron en un árbol de directorios y se sus nombres se normalizaron para automatizar su acceso.

La estructura de este árbol se explica a continuación. En el directorio 'datosTFG' encontramos una carpeta para cada usuario. En cada una de ellas encontramos carpetas para cada tipo de archivo JSON con los datos capturados:

- Entrenamiento. Contiene las actividades 'Barra' de cada entenamiento.
- ResumenSaludDiario. Contiene una carpeta por cada día de la semana de entrenamientos, llamadas 'dia1', 'dia2', ..., 'dia6'. Cada carpeta contiene los archivos JSON de Resumen de Salud de ese día.
- Estres. Contiene los archivos de estrés y batería corporal de cada día.
- Sleep. Contiene los archivos de sueño de cada día.
- Daily. Guarda archivos con un resumen general diario. De este tipo de archivos se obtiene el consumo calórico.

Estos archivos se han nombrado siguiendo una indexación por la fecha de obtención. Se ha usado este formato (y no la propia fecha) ya que los archivos de un mismo día que contaban con datos de varias pruebas se han separado en un archivo por prueba.

El formato utilizado es el siguiente:

'NombreArchivo\_XX.json', con XX = 01, 02, ..., (0)N

### 4.2.4. Análisis segmentado de los datos

En esta sección queda descrito el proceso de desarrollo del programa de análisis de los datos capturados, una vez disponemos del DataSet con los datos depurados tras el proceso de limpieza.

Como se ha comentado, se usarán los métodos y funciones de la librería Pandas (3.2.1) como la herramienta principal para el análisis y la manipulación de los datos mediante la conversión de los mismos a DataFrames (DF).

En primer lugar, explicaremos la estructura común de los bloques de análisis de cada uno de los parámetros de estudio, en la que se explica como el procedimiento de extracción y segmentación de las variables depende de la estructura original de los archivos de datos en formato JSON, poniendo ejemplos de implementación en código para cada caso.

Posteriormente, se explican las partes principales y más significativas de la implementación en código de los bloques de análisis de cada variable.

**La implementación detallada del análisis (y su justificación) del resto de variables, en la que se incluyen partes menos significativas como la declaración inicial de variables o el código para la representación gráfica y el armado de DataFrames, se incluye en el anexo B.**

#### Estructura de los bloques de análisis

En este subapartado se describe la estructura de los bloques de análisis de cada una de las variables del estudio. En siguientes secciones, se muestra la aplicación de esta estructura a cada una de dichas variables.

La estructura se divide en cuatro partes principales (gestión de directorios y carga de archivos, declaración de variables iniciales, extracción y segmentación de los datos y visualización de los resultados).

##### 1. Gestión de directorios y carga de archivos.

Siguiendo la indexación de los archivos en el árbol de directorios descrito en la sección anterior, se han definido dos funciones para la detección de estos índices y la ordenación de los directorios bajo ese criterio. Ambas funciones se explican en detalle en el anexo B:

- La función '*obtener\_numero*' usa un patrón de búsqueda con la función *search* del módulo *re*.
- La función *sorted* ordena esta lista usando *listdir* del módulo *os*, con el criterio impuesto por *obtener\_numero*.

Finalmente, para cargar los archivos, utilizamos una de las siguientes sentencias, dependiendo de la anidación de la estructura original de los archivos JSON evaluados. El procedimiento de carga se realiza dentro del bucle principal, que itera sobre cada archivo del directorio en cuestión. Ambas expresiones se han construido con debugging por si salta alguna excepción de error de lectura.

- Para los archivos con datos continuos (epoch) mapeados temporalmente, como los de frecuencia cardíaca en el entrenamiento, tiene sentido hacer una extracción organizada en filas y columnas, como DataFrames, Por tanto, para este tipo de archivos, se usa la función `read_json` de *Pandas*

Listing 4.1: Carga de archivos con Pandas

```
ruta_archivo = os.path.join(carpeta_json, archivo)

try:
    df = pd.read_json(ruta_archivo)
except ValueError as e:
    print(f"Error al leer {archivo}: {e}")
    continue
```

Listing 4.1: Carga de archivos con Pandas

- Utilizamos la función `load` del paquete `json` cuando los archivos tienen una estructura más compleja, combinando datos continuos (epoch) y discretos, lo que no tendría sentido tratar directamente como DataFrame, por lo que se cargan como diccionarios o listas de Python, que procesaremos para luego pasar los resultados a un DF. Es el caso de los archivos de Resumen de Salud, estrés, daily y sueño:

Listing 4.2: Carga de archivos con json load

```
ruta_archivo = os.path.join(carpeta_json, archivo)

try:
    with open(ruta_archivo, 'r') as f:
        data = json.load(f)
except ValueError as e:
    print(f"Error al leer {archivo}: {e}")
    continue
```

Listing 4.2: Carga de archivos con json load

2. **Declaración de variables iniciales.** Cada bloque necesitará variables, listas y diccionarios que se inicializarán antes del bucle principal de cada bloque. Estas variables, listas y parámetros se usarán para almacenar los resultados finales para su conversión a DataFrames, para el volcado manual de datos que no se encuentran en los ficheros, para hacer la distinción de días de entrenamiento y de descanso o para las etiquetas de los ejes de las gráficas.
3. **Extracción y segmentación de los datos.** Referentes al conjunto de sentencias que utilizaremos para extraer los datos requeridos y segmentarlos en variables para su manipulación.

El procedimiento y los métodos seguidos dependen de si la carga de datos fue directamente en DataFrame o en diccionarios o listas con `json.load`. También diferenciaremos si las variables a extraer son valores continuos o discretos:

- **A partir de estructuras de Python anidadas.**

Los diccionarios o listas que componen los archivos tienen una estructura muy compleja, conteniendo más listas y diccionarios como elementos, por lo que para **desanidar** estas estructuras se debe conocer su funcionamiento. Concretamente, los archivos JSON son **una lista de diccionarios, cuyos elementos pueden ser otras listas de diccionarios**.

El objetivo de el procedimiento de desanidado es el poder llegar a una estructura simple para convertirla a DataFrame, con el que luego operaremos de manera más sencilla.

A continuación se muestran ejemplos que encontramos a lo largo del código:

- **Extraer una lista de una lista de diccionarios** La siguiente sentencia extrae de la lista de datos principal de los archivos Resumen de Salud el primer elemento (`data[0]`), un diccionario, del que extraemos los campos asociados a la clave `'summaries'`, una lista que almacenamos en `summaries.list`:

```
summaries_list = data[0]['summaries']
```

Esta lista puede pasarse a DataFrame para su análisis.

Otra opción para hacer este procedimiento, en el caso de que queramos extraer varios campos del diccionario `data[0]`, es usar el método de diccionarios `get(key)`, de la siguiente forma:

```
summaries = data[0]
summaries_list = summaries.get('summaries', {})
```

- **Extracción de un elemento discreto de un diccionario.** También podemos usar el método `get` para extraer un elemento discreto de un diccionario añadiendo el índice en el que se encuentra. Por ejemplo, las calorías en reposo se encuentran en el campo `bmrKilocalories` de un diccionario en la lista principal. La extracción es la siguiente:

```
calorias_data = data[0]
bmr_calories = calorias_data.get('bmrKilocalories',
                                  0)
```

- **A partir de DataFrames.** Una vez tenemos los datos organizados como DataFrames, se usarán los siguientes métodos para su extracción y segmentación.

- **Extraer todos los datos con el mismo campo de una columna.** Es el caso de la frecuencia cardíaca durante el entrenamiento en los archivos *Entrenamiento*. Estos valores se encuentran en la lista de diccionarios `'samples'` que contienen información de varias variables por cada marca temporal. Esta estructura se pasa a un DataFrame donde la columna `'samples'` contiene una lista con cada uno de los diccionarios de datos por muestra temporal.

En primer lugar, se usa la función `apply` para quedarnos únicamente con el elemento `'hearRate'` de cada diccionario de la columna, resultando una fila con todos los valores de FC. Almacenamos este resultado en una nueva columna del DF, por lo que nos queda una lista con el valor de FC asociado a cada fila.

Finalmente, extraemos los elementos de esta columna a una lista de Python, usando el método `iloc[0]` para extraer el primer elemento de la columna.

```
df['heartRate'] = df['samples'].apply(lambda x: [
    item['heartRate'] for item in x])

heart_rate_flat = df['heartRate'].iloc[0]
```

- **Extracción de un valor discreto.** Para acceder a un valor concreto, simplemente usamos la función `loc` de *Pandas*, indicando que se acceda a la fila (valor) deseada de una columna que cumpla la condición de búsqueda. El siguiente ejemplo lo ilustra: El valor de respiración media de un resumen de salud se encuentra

en la columna '*avgValue*', en la fila en la que el valor de columna '*summaryType*' es '*respiration*', por lo que usaremos *loc* de esta manera:

```
respiracion = summaries_df.loc[summaries_df['  
summaryType'] == 'respiration', 'avgValue'].  
values[0]
```

4. **Visualización de los resultados.** Referente al procesado de los DataFrames de resultados y al conjunto de sentencias para la representación gráfica de las variables para cada día del estudio, que incluiremos dentro del bucle principal, y para su evolución semanal, al final de cada bloque.

Como se ha comentado, la implementación de esta parte se recoge en el apéndice B para la mayoría de las variables.

### Frecuencia cardíaca durante el entrenamiento de boxeo

En este apartado se recoge el proceso de análisis de los datos de frecuencia cardíaca capturados durante las sesiones de entrenamiento y almacenados en los archivos JSON del directorio 'Entrenamiento'. Estos archivos cuentan con un resumen con los valores significativos (máximo, medio, ...) y un mapeo temporal de los valores de FC a lo largo del periodo de la actividad.

El objetivo es, para cada día, extraer este mapeo de FC a lo largo del tiempo de entrenamiento y realizar una representación gráfica del mismo. Además, también se registran y representan los valores máximo y mínimo, y se traza una recta con el valor medio de cada sesión.

Las variables inicializadas para ello corresponden a listas con el nombre de los días (para la columna del DataFrame que asocia los valores a su día de la semana y que será el parámetro común de todos los DF), una lista con los días de entrenamiento, una lista para almacenar los resultados y un contador para asignar el número de entrenamiento al título de las gráficas.

El bucle principal, en el que se realiza la extracción de los datos siguiendo el procedimiento de extracción de datos continuos almacenados en listas con *Pandas* ya comentado, el procesado para su disposición gráfica y dicha representación, es el siguiente:

Listing 4.3: Bucle de extraccion de datos de frecuencia cardiaca

```
#Bucle que itera sobre los archivos  
for archivo in archivos_json:  
  
    ##Carga de Archivos##  
  
    #Extraccion (y verificacion) de datos  
    df['heartRate'] = df['samples'].apply(lambda x: [item['heartRate'] for item  
        in x])  
    df['averageHeartRateInBeatsPerMinute'] = df['summary'].apply(lambda x: x['  
        averageHeartRateInBeatsPerMinute'])  
  
    if df['heartRate'].isnull().all() or df['averageHeartRateInBeatsPerMinute'  
        ].isnull().all():  
        print(f"Datos de frecuencia cardiaca faltantes en el archivo {archivo}.")  
        continue  
  
    #Preparamos valores para la grafica y los almacenamos en un array
```

```

heart_rate_flat = df['heartRate'].iloc[0]
hr_medio = df['averageHeartRateInBeatsPerMinute'].mean()
mean_hr = [hr_medio] * len(heart_rates_flat) #Se crea una lista a lo largo
    del tiempo para imprimirla en la grafica
max_hr = max(heart_rates_flat)
max_index = heart_rates_flat.index(max_hr)
min_hr = min(heart_rates_flat[300:]) #Ignorar los primeros valores para
    encontrar el minimo
min_index = heart_rates_flat.index(min_hr, 300)

hr_values.append({
    'Dia': dias_entrenamiento[contador_entrenamiento],
    'hr_medio': hr_medio,
    'max_hr': max_hr,
    'min_hr': min_hr
})

## Representacion grafica de la evolucion diaria de los parametros de FC ##

```

Listing 4.3: Bucle de extraccion de datos de frecuencia cardiaca

Finalmente, creamos un DataFrame con estos valores ordenados por día y realizamos una representación de la evolución de los valores medios, máximos y mínimos a lo largo de la semana. Para la creación del DF, añadimos los días (añadimos con *concat*, y ordenamos los índices categóricos de días de la semana con *Categorical* y *sort\_values*) para los que no hay entrenamiento (con valores nulos), por lo que para la representación tendremos que hacer un filtrado (con *dropna*) para quedarnos con los días en los que si hubo sesión de boxeo.

### Análisis de picos y valles de frecuencia cardíaca durante el entrenamiento

A continuación, se muestra el proceso de filtrado de valores para diferenciar la frecuencia cardíaca media de los asaltos de actividad de la de los asaltos de recuperación. El procedimiento consiste en la detección de una serie de picos y de valles con la función *find\_peaks*, a la cual se le realizará un primer proceso de filtrado mediante valores umbral. Se realizará un segundo filtrado manual para los valores no deseados que pasen el filtro anterior. Finalmente, se calcularán los valores medios y se guardarán.

Entre las variables iniciales, se incluyen un diccionario, para cada archivo, de listas de índices de picos y valles. Es parte del proceso de filtrado manual de máximos y mínimos locales no deseados, de manera posterior al filtrado automático. Los índices de los picos detectados se imprimen con las gráficas, por lo que el analista puede localizar los de los no deseados y añadirlos a estas listas de borrado.

Listing 4.4: Diccionarios para el filtrado manual de picos y valles por su índice

```

#Diccionarios, listas y variables iniciales.
#Especificamos los picos y valles a filtrar manualmente para cada archivo
    segun su indice (impreso mas adelante)
picos_a_eliminar_manual = {
    'EntrenamientoBoxeo_1.json': [3222, 5714, 2702],
    'EntrenamientoBoxeo_2.json': [2871, 3578, 5740],
    'EntrenamientoBoxeo_3.json': [],
    'EntrenamientoBoxeo_4.json': [4319]
}

```

```

}

valles_a_eliminar_manual = {
    'EntrenamientoBoxeo_1.json': [1, 353, 609, 844, 1909, 2984, 4215],
    'EntrenamientoBoxeo_2.json': [7, 441, 607, 2534, 3717],
    'EntrenamientoBoxeo_3.json': [3, 274, 552, 793, 1840, 2623, 3584, 3810,
        4122, 4324],
    'EntrenamientoBoxeo_4.json': [29, 399, 757, 1046, 1479, 1843, 3439]
}

```

Listing 4.4: Diccionarios para el filtrado manual de picos y valles por su índice

En el bucle principal de FC durante el entrenamiento, añadimos la detección de máximos y mínimos locales, el filtrado por umbral y el filtrado manual. Finalmente, realizamos la visualización gráfica de los picos (y su listado para el filtrado manual) y su valor medio para cada entrenamiento.

En la función `find_peaks` introducimos como parámetro la distancia mínima entre picos, para que se ajuste a las características temporales de nuestras gráficas.

Por otro lado, el filtrado automático se realiza con un umbral ubicado en la media de FC total con un pequeño margen superior e inferior. Por tanto nos quedaremos con los picos superiores a la media - margen y con los valles inferiores a la media + margen.

Finalmente, el procesado manual se realiza con una estructura condicional que selecciona los picos y valles si su índice no se encuentran en la lista de picos y valles a eliminar.

La última parte del bucle corresponde a la representación de los picos y valles (y su valor medio) sobre la gráfica de frecuencia cardíaca de cada entrenamiento, véase el apéndice B. Nos apoyamos de esta gráfica para ver qué picos o valles eliminar manualmente.

Listing 4.5: Bucle de extraccion de datos de picos de frecuencia cardiaca

```

#Deteccion de picos y valles
picos, _ = find_peaks(heart_rates_flat, distance=100)
valles, _ = find_peaks([-hr for hr in heart_rates_flat], distance=100)

#Filtrado de picos segun un umbral: margen alrededor de la media
media_hr = pd.Series(heart_rates_flat).mean()
picos = [p for p in picos if heart_rates_flat[p] > media_hr - 3]
valles = [v for v in valles if heart_rates_flat[v] < media_hr + 3]

#Filtrado de picos manual
if archivo in picos_a_eliminar_manual:
    picos = [p for p in picos if p not in picos_a_eliminar_manual[archivo]]
if archivo in valles_a_eliminar_manual:
    valles = [v for v in valles if v not in valles_a_eliminar_manual[
        archivo]]

#Calculamos los valores medios y los almacenamos
if picos:
    picos_media = pd.Series([heart_rates_flat[p] for p in picos]).mean()
    picos_medios.append(picos_media)
if valles:
    valles_media = pd.Series([heart_rates_flat[v] for v in valles]).mean()
    valles_medios.append(valles_media)

```

```
#Imprimimos la lista de picos para el filtrado manual
print(f"Indices de picos detectados en {archivo}: {picos}")
print(f"Indices de valles detectados en {archivo}: {valles}")

## Grafica de picos y valles por entrenamiento ##
contador_entrenamiento += 1
```

Listing 4.5: Bucle de extraccion de datos de picos de frecuencia cardiaca

Acabamos creando un DataFrame con los valores medios de picos y valles, que corresponden a la frecuencia cardíaca de los asaltos de actividad y recuperación, respectivamente, y representando la evolución de estos valores medios a lo largo de la semana (Anexo B).

### Variabilidad de la frecuencia cardíaca diaria

Los valores de HRV los encontramos en los archivos de Resumen de SALUD, almacenados en carpetas por día (dia1,..., dia6) en el directorio 'ResumenSaludDiario'. De estos archivos únicamente (para este parámetro) nos quedamos con el valor de HRV RMSSD, almacenado en el campo 'avgValue' con nombre 'rmssd\_hrv'.

Entre las variables iniciales, encontramos una lista con los nombres de las carpetas 'días', la lista con los días de la semana para el campo común del DF de resultados, listas para ajustar los ejes y colores de las gráficas y la lista para almacenar los resultados.

En el bucle principal extraeremos y representaremos (B) los datos de HRV RMSSD de cada día según el momento en el que se obtuvieron. Para entender la evolución lineal del parámetro, la espaciación del eje X se hace según las horas del día, asignando una hora a cada momento del día. También se ha hecho la diferenciación entre días de entrenamiento, en los que hay 5 tomas, y de descanso, en los que solo hay 3, ajustando los momentos del día de manera apropiada.

En este archivo, la iteración del bucle principal se produce sobre las carpetas de los diferentes días, y se produce una segunda iteración para la carga de archivos con otro bucle for sobre los archivos, como habíamos hecho anteriormente. Por tanto, la función de ordenación de archivos se produce dentro del bucle, ya que se ha de hacer para cada día.

La segmentación de los datos sigue la estructura comentada de extracción de una lista ('summaries\_list') de datos asociados a la misma clave ('summaries') en una lista de diccionarios. Esta lista se convierte en un DataFrame, y accedemos al campo 'avgValue' con el método 'loc' usando como condicional de búsqueda de la fila que el valor de la columna 'summaryType' sea 'rmssd\_hrv', como se comentó anteriormente.

Listing 4.6: Bucle de extraccion y representacion de datos de HRV diario

```
for idx, dia in enumerate(dias):
    carpeta_dia = os.path.join(carpeta_principal, dia)
    ##Funcion de ordenacion de archivos para cada dia##

    # Lista para almacenar los valores de RMSSD para cada archivo
    rmssd_values = []
    horas_actuales = []

    #Diferenciamos entre dias con y sin entrenamiento.
    if dia in ['dia3', 'dia6']: #Sin
        horas_dia = [8, 14, 24] #Solo 8 AM, 2 PM y 12 AM
```

```

archivos_json = archivos_json[:3]
else:
    horas_dia = horas_x #Con
    archivos_json = archivos_json[:5]

#Bucle de carga de archivos
for archivo, hora in zip(archivos_json, horas_dia): #Asignamos las horas
    correspondientes a cada archivo
    ruta_archivo = os.path.join(carpeta_dia, archivo)

    with open(ruta_archivo, 'r') as f:
        data = json.load(f)

    summaries_list = data[0]['summaries'] #Accede al primer objeto del
    diccionario y almacena la lista
    summaries_df = pd.DataFrame(summaries_list)

    rmssd_value = summaries_df.loc[summaries_df['summaryType'] == ,
        'rmssd_hrv', 'avgValue'].values[0]
    rmssd_values.append(rmssd_value)
    horas_actuales.append(hora)

    todos_rmssd_values.append(rmssd_values)
    todas_horas_x.append(horas_actuales)

## Grafica de representacion diaria del HRV ##

```

Listing 4.6: Bucle de extraccion y representacion de datos de HRV diario

Finalmente, creamos el DataFrame con los valores de HRV diarios a lo largo de la semana para hacer una representación conjunta, B.

### Frecuencia respiratoria y niveles de estrés pre y post entrenamiento

De los archivos de Resumen de Salud realizados antes y después del entrenamiento también obtenemos información valiosa, como la respiración media, almacenada en 'avgValue' del campo 'respiration', y el nivel de estrés medio, almacenado también en el 'avgValue' del campo 'stress'.

Para la respiración, representaremos los valores medios pre y post entrenamiento, junto al del propio entrenamiento, que volcaremos manualmente en un diccionario a partir de la información de la actividad en Garmin Connect.

Para el estrés, representaremos únicamente los valores medios pre y post entrenamiento, para poder apreciar su diferencia.

Entre las variables declaradas, destaca el diccionario *entrenamientos* que recoge días son de entrenamiento y para cada uno de esos días, una tupla que indica qué archivos (según su índice) de Resumen de Salud corresponden al pre y post entrenamiento, respectivamente. Por otro lado, los valores de respiración media durante el entrenamiento se definen en otro diccionario.

El bucle principal sigue una estructura diferente, precisamente por la necesidad de diferenciar qué archivos corresponden a pre entrenamiento y cuáles a post entrenamiento. Destaca el distinto enfoque que le damos a la carga de los datos, ya se ha hecho mediante una función, *cargar\_datos\_pre\_post*, que devuelve el par de valores de datos de estrés y respiración según

el nombre del archivo que se le pase de parámetro de entrada. Se usa este enfoque ya que nos permite hacer la distinción pre/post indicando explícitamente el nombre de los archivos correspondientes a cada periodo según el día evaluado, usando el diccionario *entrenamientos*.

Los índices de dicho diccionario se almacenan en las variables *pre\_entrenamiento\_idx* y *post\_entrenamiento\_idx*, con los que se crea la lista *archivos\_necesarios* normalizada a nuestra indexación, para finalmente declarar los *archivos\_presentes* de nuestro DataSet que acaben en esos índices, tras un barra baja y con la extensión *.json*.

Listing 4.7: Bucle de extracción de datos de estrés y respiración pre/post entrenamiento

```
#Bucle principal
for idx, dia in enumerate(dias):

    ...

    if dia in entrenamientos: #Los dias de entrenamiento dividimos entre pre y
        post_entrenamiento
        pre_entrenamiento_idx, post_entrenamiento_idx = entrenamientos[dia] #Se
            declara para dicho dia los archivos pre y post segun el diccionario

        ## Carga de datos ## Se indica que archivos de la lista buscamos, para
            cargar los datos diferenciando entre pre y post
        archivos_necesarios = [f'{pre_entrenamiento_idx:02d}', f'{{
            post_entrenamiento_idx:02d}}']
        archivos_presentes = [archivo for archivo in archivos_json if any(f'_{{
            num}}.json' in archivo for num in archivos_necesarios)]

        #Funcion que carga dos datos de estres y respiracion segun nombre del
            archivo, nos permite cargar diferenciando entre pre y post
        def cargar_datos_pre_post(archivo):
            with open(os.path.join(carpeta_dia, archivo), 'r') as f:
                data = json.load(f)
            if isinstance(data, list) and 'summaries' in data[0]:
                summaries_df = pd.DataFrame(data[0]['summaries'])
                respiracion = summaries_df.loc[summaries_df['summaryType'] == 'respiration', 'avgValue'].values[0]
                estres = summaries_df.loc[summaries_df['summaryType'] == 'stress', 'avgValue'].values[0]
                return respiracion, estres
            else:
                print(f"Advertencia: No se encontraron datos de 'summaries' en {{
                    archivo}}")
                return None, None

            respiracion_pre_value, estres_pre_value = cargar_datos_pre_post(
                archivos_presentes[0])
            respiracion_post_value, estres_post_value = cargar_datos_pre_post(
                archivos_presentes[1])

    ...
```

Listing 4.7: Bucle de extracción de datos de estrés y respiración pre/post entrenamiento

Finalmente se almacenan en diccionarios para crear los DataFrames resultantes, haciendo la gestión de días sin entrenamiento.

### Nivel de estrés diario y batería corporal

Ambos parámetros se encuentran en los archivos 'stress' almacenados, tras su procesado, en la carpeta 'Estrés'.

El objetivo es poder visualizar el impacto del nivel de estrés en la bajada o subida de la batería corporal, por lo que los representaremos de manera conjunta a lo largo del día.

Entre las variables iniciales destaca que en este caso definimos los días de entrenamiento como una lista con sus índices ya que realizamos cálculos en el bucle para cada uno de ellos (en bloques anteriores, usábamos el nombre del día ya que directamente añadíamos los días sin entrenamiento al DF con valores nulos). Por otro lado, incluimos el diccionario *entrenamientos* para introducir manualmente la hora de inicio y la duración de los entrenamientos de la semana:

Listing 4.8: Diccionario para definir el periodo de entrenamiento para los archivos de estrés y Body Battery

```
#Diccionarios y listas para almacenar variables
entrenamientos = { #Hora inicio y duracion de los entrenamientos
    "Lunes": ("17:37", "01:26"),
    "Martes": ("17:33", "01:15"),
    "Jueves": ("17:57", "01:15"),
    "Viernes": ("17:35", "01:21"),
}
```

Listing 4.8: Diccionario para definir el periodo de entrenamiento para los archivos de estrés y Body Battery

En el bucle principal, que itera sobre los archivos de la carpeta principal, se realizan varias funciones. Tenderemos en cuenta que los instantes de tiempo en los que el sensor no pudo trackear el estrés se señalan con valores nulos.

En primer lugar inicializamos las variables que guardaremos en los DataFrame con valores nulos, ya que se comprobó que los datos estaban corruptos para uno de los usuarios, por lo que debemos asignar valores para poder construir el DF. Seguidamente, se extraen los datos y se crea un DF para cada variable y su respectivo mapeo temporal (en horas). Se filtran los valores nulos, de valor negativo. A partir de DF de batería corporal, se calcula el valor máximo (obtenido al levantarse el usuario) y con el que acaba el día (usando *iloc[-1]* para acceder al último elemento de la columna), y sus respectivos tiempos. Estos valores se almacenan en una lista para crear un DF final.

A continuación, se muestra y comenta la parte más interesante de este bloque de análisis, correspondiente a la distinción entre los valores de estrés registrados de manera previa y posterior al entrenamiento.

Para ello, se usa el diccionario *entrenamientos* y la función *convertir\_a\_segundos* que usamos en el depurado de datos, almacenando los valores de inicio y final del entreno en horas en dos variables. Almacenamos el valor de la batería corporal previo al entrenamiento y lo añadimos a la lista de BB.

Se calcula el valor medio de estrés pre-entreno creando un DF con los valores registrados desde que comienza el día (marca temporal en la que se alcanza el máximo de BB) y el inicio del entrenamiento y calculando su media. El valor post-entreno será la media del DF con los valores de estrés desde que acaba el entrenamiento hasta el final del día (instante en el que se alcanza

el último valor de BB). El valor de BB antes del entrenamiento también se extrae (con `iloc[-1]`, ya que es el último valor de la columna '`BodyBattery`' del DF de valores pre entrenamiento), y se almacena en la lista.

Por último, se hace la gestión de datos para los días de descanso.

Listing 4.9: Distinción pre/post entrenamiento en el bucle de extraccion y representacion de datos de estres y Body Battery

```
## Hacemos separacion pre y post entrenamiento para el estres ##
#Detectar si es un dia de entrenamiento
#Detectar si es un dia de entrenamiento y calcular inicio y fin
if dia_semana in entrenamientos:
    hora_inicio, duracion = entrenamientos[dia_semana]
    inicio_entrenamiento = convertir_a_segundos(hora_inicio) / 3600
    fin_entrenamiento = inicio_entrenamiento + convertir_a_segundos(
        duracion) / 3600

    #BB pre entrenamiento
    df_pre_entrenamiento_battery = df_body_battery[df_body_battery[',
        TimeOffset'] < inicio_entrenamiento]
    if not df_pre_entrenamiento_battery.empty:
        battery_pre_entrenamiento = df_pre_entrenamiento_battery[',
            BodyBattery'].iloc[-1]

    #Estres pre y post entrenamiento
    df_pre_entrenamiento = df_stress[(df_stress['TimeOffset'] >=
        max_battery_time) &
        (df_stress['TimeOffset'] <
            inicio_entrenamiento) &
        (df_stress['StressLevel'] >= 0)]
    df_post_entrenamiento = df_stress[(df_stress['TimeOffset'] >
        fin_entrenamiento) &
        (df_stress['StressLevel'] >= 0)]

    estres_pre = df_pre_entrenamiento['StressLevel'].mean() if not
        df_pre_entrenamiento.empty else None
    estres_post = df_post_entrenamiento['StressLevel'].mean() if not
        df_post_entrenamiento.empty else None

    estres_dias.append({
        'Dia': dia_semana,
        'Estres_Pre-Entrenamiento': estres_pre,
        'Estres_Post-Entrenamiento': estres_post
    })

    #Anadimos BB pre a la lista
    body_battery_dias[-1]['Body_Battery_Pre-Entrenamiento'] =
        battery_pre_entrenamiento
else:
    #Dias de descanso
    df_dia_descanso = df_stress[df_stress['TimeOffset'] >=
        max_battery_time]
```

```
estres_medio_descanso = df_dia_descanso['StressLevel'].mean() if  
    not df_dia_descanso.empty else None  
  
estres_dias.append({  
    'Dia': dia_semana,  
    'Estres_Medio_Dia_Descanso': estres_medio_descanso  
})
```

Listing 4.9: Distinción pre/post entrenamiento en el bucle de extraccion y representacion de datos de estres y Body Battery

La última parte del bucle corresponde a la confección de la gráfica, véase el anexo B. Finalmente, se crean ambos DataFrames.

### Balance calórico por tipo de actividad

Este parámetro se obtiene de los archivos en la carpeta 'Daily'. En el parámetro 'bmrKilocalories' de estos archivos se encuentran las consumidas en reposo y en 'activeKilocalories' las consumidas de manera activa. Las consumidas en el entrenamiento se introducen de manera manual en un diccionario, ya que Garmin no las brinda en los archivos JSON de esa actividad. Para todos los tipos de calorías, se inicializa una lista.

Las variables y listas iniciales definidas son las siguientes:

Listing 4.10: Diccionario con el gasto calórico de cada entrenamiento

```
#Volcado manual de las calorías de los entrenamientos  
calorias_entrenamiento = { # 1:Lunes, 2:Martes, 4:Jueves, 5:Viernes  
    1: 1266,  
    2: 1196,  
    4: 1025,  
    5: 1043  
}
```

Listing 4.10: Diccionario con el gasto calórico de cada entrenamiento

El bucle para extraer y almacenar los datos de consumo calórico por día y tipo de ejercicio (anexo B) simplemente sigue el método de extracción de un valor discreto en un diccionario con la función `get()`, descrito en la sección de la arquitectura de los bloques de análisis.

Finalmente, creamos un DataFrame con los datos de consumo calórico y representamos gráficamente este gasto diario por tipo de actividad, con un diagramas de barras superpuestas (anexo B).

### División en zonas y calificación del periodo de sueño

En este apartado se recoge el proceso de análisis de las jornadas de sueño de cada día del estudio. Se visualizará de manera gráfica el paso por las distintas zonas de sueño a lo largo de la noche y se mostrará la calificación de cada día según el sistema de rúbricas de Garmin.

Estos datos se encuentran en los archivos JSON en la carpeta Sleep.

En primer lugar, se construyó una función para el procesado de las zonas de sueño (que divide el eje *Y* de la gráfica en 3 alturas según el tipo de zona de sueño: ligero-1, profundo-2, REM-3)y otra para la asignación de colores a cada calificación (que volveremos a usar en la

implementación del algoritmo de calificación), de cara a la construcción de una gráfica de zonas de sueño a lo largo del tiempo y una tabla de calificaciones, respectivamente:

Listing 4.11: Funciones de procesado para las graficas de sueño

```
#Funcion de gestion de niveles del sueno. Prepara los ejes
def procesar_fases_sueno(sleep_levels_map):
    tiempos = []
    niveles = []
    for nivel, segmentos in sleep_levels_map.items():
        if nivel == "unmeasurable": # Ignorar segmentos 'no medibles'
            continue
        altura = 0
        if nivel == "light":
            altura = 1 # Altura s. ligero
        elif nivel == "deep":
            altura = 2 # Altura s. profundo
        elif nivel == "rem":
            altura = 3 # Altura REM

        for segmento in segmentos:
            tiempos.append((segmento['startTimeInSeconds'], segmento[,
                'endTimeInSeconds']))
            niveles.append(altura)

    return tiempos, niveles

#Asignacion de colores segun la calificacion
def color_cell(value):
    if value == 'POOR':
        return 'background-color: red'
    elif value == 'FAIR':
        return 'background-color: yellow'
    elif value == 'GOOD':
        return 'background-color: lightgreen'
    elif value == 'EXCELLENT':
        return 'background-color: green'
    return ''
```

Listing 4.11: Funciones de procesado para las graficas de sueño

El bucle principal (anexo B) itera sobre los archivos y sobre un índice para ajustar los días de la semana. Dentro del mismo se realiza la representación gráfica de los niveles de sueño a lo largo del tiempo. Para ello, se hace el mapeo zona de sueño - tiempo con la función *procesar\_fases\_sueno*, en la que se le da la altura en el eje *Y* para la representación gráfica.

Por otro lado, se extraen las calificaciones de cada aspecto de la rúbrica de sueño que utiliza Garmin (con el método de extracción de valores discretos en diccionarios, usando la función *get()*), y se almacena en la lista *calificaciones\_dias* definida previamente.

Finalmente, se almacenan las calificaciones de cada día en un DataFrame y se usa el método *style* de *Pandas* para visualizarlo con el mapeo de colores de la función *color\_cell*.

#### 4.2.5. Aplicación del algoritmo de calificación de la recuperación y el rendimiento

En este apartado se describe la estructura del código empleado para la implementación del algoritmo de calificación del rendimiento deportivo y la recuperación descrito en la sección 3.3.

El objetivo es presentar al usuario una visualización intuitiva que le permita relacionar su recuperación con su desempeño en el entrenamiento, mostrando la calificación global de ambos aspectos desglosada en la calificación de cada uno de los parámetros que componen el análisis. Además, para cada día, mediante un sistema de desplegables, recibirá un conjunto de comentarios o insights que contengan el valor registrado y que justifiquen la calificación y/o brinden recomendaciones. El valor numérico se incluye en el comentario y no en la tabla para simplificar la interfaz, teniendo en cuenta que el usuario final querrá información que pueda entender, sin necesidad de ser experto en ciencias del deporte para usar el sistema. En caso de querer saber el valor registrado, simplemente accederá a los comentarios.

La implementación comienza con **funciones de evaluación** que reciben como parámetro de entrada los DataFrames resultantes de la fase de análisis con los datos ordenados por día, y brindan un nuevo DataFrame, compuesto por una **calificación** (según el sistema de cuatro niveles) comparando los valores de entrada con los de referencia, como se recogía en las tablas 3.6 y 3.10, y un **insight** definido en un conjunto de diccionarios que clasifican estos comentarios por parámetro y calificación (véase el Apéndice C).

Estos nuevos DataFrames y los pesos de cada parámetro, serán una de las entradas de las **funciones de calificación de recuperación y rendimiento**, que extrae la calificación de cada parámetro para cada día y calcula la puntuación global con el método descrito en la sección 3.3. Con esta calificación y los insights de cada parámetro recopilados por día se arma un DF final.

Finalmente, se juntan ambos DFs por día y se imprime con un código de colores. Los insights se muestran con un sistema de desplegables por día.

#### Funciones de evaluación

A continuación se comenta la estructura de las funciones de evaluación y calificación de los parámetros. Para ello, usaremos de ejemplo la función empleada para la calificación de los parámetros de frecuencia cardíaca durante el entrenamiento, *evaluar\_fc*. El resto de funciones, de igual estructura, se incluyen en el apéndice C.

Esta estructura se compone fundamentalmente de un bucle principal, que itera con el método de *iterrows()* de *Pandas* recorriendo las filas de cada columna y su índice, además de las siguientes partes:

- **Preparación del DataFrame.** De manera previa a la función, se ensambla el DF que será evaluado, mediante un pequeño procesamiento. Para ello, se usa el siguiente conjunto de métodos de Pandas para el dimensionado y procesado de DFs, explicado en la sección 3.2.1.

- *merge(DF1, DF2, on, how)*. Para aunar DataFrames según parámetros comunes. En concreto, usaremos la columna *on = Día* como parámetro común y el método *how=outer*, que incluye todos los valores de la columna común aunque en alguno de los DF su valor sea nulo.

Es el caso de la confección del DF para la evaluación de frecuencia cardíaca, donde se juntan los DFs de FC media y de picos y valles resultantes del análisis.

- *loc[x, y]*. Para manipular el valor concreto de una posición del DF.
- *shift(i)*. Para desplazar las filas de una columna *i* veces.

Lo usaremos para generar nuevas columnas para los valores de datos del '**día anterior**', como el estrés post entrenamiento o el body battery final. Así, colocaremos en la fila del martes los valores medidos el lunes. Esto nos permite simplificar los cálculos de cada día.

Para usar este parámetro, debemos usar una copia del DF original, que generaremos con el método *copy()*, ya que si no salta una excepción.

- **Parámetros de entrada.** Estas funciones recogen como parámetros de entrada los DataFrames con los datos a evaluar, el diccionario con insights sobre los mismos y los valores de referencia que usaremos de umbral.
- **Declaración de variables y extracción de los datos.** Se declara una lista para los resultados.  
La extracción de datos se realiza almacenando el valor de cada una de las filas iteradas con *iterrows()* en una variable.
- **Evaluación.** Para la evaluación de cada parámetro, se usa una estructura condicional que asigna un nivel (de la estructura de calificación de cuatro niveles) en una variable *calificacion* según los rangos o umbrales de referencia establecidos en las tablas 3.6 y 3.10. Además, se almacenará el insight con el comentario especificado en el diccionario introducido como entrada, correspondiente a dicha calificación para dicho parámetro, junto al valor numérico del mismo (con sus unidades).
- **Resultados.** En cada iteración, calificación y comentario se añaden a un diccionario que los asocia al día correspondiente. Estos valores son unidos a la lista *resultados[]*.  
Finalmente, la función devuelve un DataFrame creado a partir de la lista de resultados generada, nombrados como *resultados\_<parámetro>*

A continuación, se muestra el código de la función *evaluar\_fc*, que sigue la estructura descrita:

Listing 4.12: Función de evaluación de los parámetros de FC

```
#DF con datos de FC
fc_df = pd.merge(hr_df, df_picos_valles, on='Dia', how='outer')

#Funcion de evaluacion y calificacion de los parametros de frecuencia cardiaca
#durante el entrenamiento
def evaluar_fc(df, insights_dict, zona_FC_optima, FC_asalto, FC_recuperacion):
    resultados = []

    for index, row in df.iterrows():
        #Extraccion de los datos
        dia = row['Dia']
        hr_medio = row['hr_medio']
        picos_medios = row['Picos_Medios']
        valles_medios = row['Valles_Medios']

        #Evaluacion de la FC media con ZFC0
        if pd.notna(hr_medio):
```

```

if zona_FC_optima[0] + 5 <= hr_medio <= zona_FC_optima[1] - 5:
    calificacion_hr = 'Excelente'
elif zona_FC_optima[0] <= hr_medio <= zona_FC_optima[1]:
    calificacion_hr = 'Buena'
elif (zona_FC_optima[0] - 5 <= hr_medio < zona_FC_optima[0]) or (
    zona_FC_optima[1] < hr_medio <= zona_FC_optima[1] + 5):
    calificacion_hr = 'Regular'
else:
    calificacion_hr = 'Mala'
else:
    calificacion_hr = 'Mala'
comentario_hr = f"{insights_dict[calificacion_hr]['hr_medio']} (Valor:{hr_medio:.2f} bpm)"

#Evaluacion de la FC de asaltos de actividad
FC_ref = max(FC_asalto, zona_FC_optima[1])
if pd.notna(picos_medios):
    diff_picos = FC_ref - picos_medios
    if diff_picos > 5:
        calificacion_picos = 'Excelente'
    elif -2 < diff_picos <= 5:
        calificacion_picos = 'Buena'
    elif -10 <= diff_picos <= -2:
        calificacion_picos = 'Regular'
    else:
        calificacion_picos = 'Mala'
else:
    calificacion_picos = 'Mala'
comentario_picos = f"{insights_dict[calificacion_picos]['Picos_Medios']} (Valor:{picos_medios:.2f} bpm)"

#Evalaucion de la FC de asaltos de recuperacion
if pd.notna(valles_medios):
    diff_valles = FC_recuperacion - valles_medios
    if diff_valles > 5:
        calificacion_valles = 'Excelente'
    elif -2 < diff_valles <= 5:
        calificacion_valles = 'Buena'
    elif -10 <= diff_valles <= -2:
        calificacion_valles = 'Regular'
    else:
        calificacion_valles = 'Mala'
else:
    calificacion_valles = 'Mala'
comentario_valles = f"{insights_dict[calificacion_valles]['Valles_Medios']} (Valor:{valles_medios:.2f} bpm)"

#Creamos diccionario con calificaciones e insights
resultado_dia = {
    'Dia': dia,
    'Calificacion_hr_medio': calificacion_hr,
}

```

```

    'Insight_hr_medio': comentario_hr,
    'Calificacion_Picos_Medios': calificacion_picos,
    'Insight_Picos_Medios': comentario_picos,
    'Calificacion_Valles_Medios': calificacion_valles,
    'Insight_Valles_Medios': comentario_valles
}
resultados.append(resultado_dia)

#Devolvemos un DF con los
return pd.DataFrame(resultados)
#Llamamos a la función y almacenamos el DF
resultados_fc = evaluar_fc(fc_df, insights_dict_fc, zona_FC_optima, FC_asalto,
                           FC_recuperacion)

```

Listing 4.12: Función de evaluación de los parámetros de FC

## Funciones de calificación

En este apartado se describe el funcionamiento de las funciones de calificación de la recuperación y el rendimiento en el entrenamiento, *calificar\_recuperacion* y *calificar\_rendimiento*. Ambas funciones tienen una estructura análoga, con algunas adaptaciones.

Esta estructura es la descrita a continuación:

- **Parámetros de entrada.** Las entradas de ambas funciones son, fundamentalmente, los DataFrames de salida de las funciones de evaluación de cada parámetro y los diccionarios con sus pesos de acuerdo al sistema de evaluación (siguiendo el sistema de evaluación descrito en la sección 3.3). Para ello, se han definido cuatro diccionarios de pesos diferenciados: tres para la calificación de la recuperación, según la distinción por tipo de día, y uno para el rendimiento.

Por otro lado y como se comenta en la sección de resultados, los archivos de estrés y body battery para uno de los usuarios presentaban datos no válidos, por lo que en la función de recuperación se introduce como entrada una variable booleana, *valid\_stress* para indicar si los datos de estrés y BB son válidos o no.

- **Declaración de variables y extracción de los datos.** De nuevo, se declara una lista *resultados* que se devolverá como salida de ambas funciones.

El bucle principal de la función itera con un índice sobre los días de la semana.

En ambos casos, se hace una **distinción entre los tipos de días** posibles para la **extracción de los datos**:

- **Función de recuperación.** Según el sistema de calificación descrito en la sección 3.3, mediante una estructura condicional que chequea el nombre del día iterado variaremos el diccionario de *parámetros* a tener en cuenta según si el día en cuestión es de descanso, de entrenamiento tras descanso o de entrenamiento tras entrenamiento. Además, con la variable booleana *valid\_stress* controlamos si añadimos al diccionario los parámetros de estrés y body battery correspondientes al análisis de cada tipo de día (el body battery inicial es válido en todos los casos).

Para cada parámetro incluido en el diccionario, se ha extraído su calificación y el insight que la acompaña.

Finalmente y según el tipo de día, seleccionamos el diccionario de pesos a utilizar en el cálculo de la puntuación.

- **Función de rendimiento.** Como la calificación del rendimiento solo procede en los días de entrenamiento, tenemos que separarlos de los días en los que hay descanso.

Para ello, introducimos un bloque *if-continue*, en el que directamente confeccionamos el diccionario de resultados diarios para los días de descanso declarando todos los parámetros con valor nulo. Se unirán estos resultados a la lista de resultados finales y la sentencia *continue* saltará el bloque para el resto de días.

Para el resto de días, se declara el diccionario *parametros* en el que se hace la extracción de datos de las calificaciones y los insights de cada variable a partir de los DFs de entrada.

Por otro lado, se almacena la calificación del VO<sub>2</sub> máximo del usuario para la posterior adaptación de los límites de calificación.

- **Sistema de calificación.** El cálculo de la puntuación de ambas variables es análogo y sigue el sistema descrito en la sección 3.3.

Para su implementación, hemos inicializado variables para la puntuación total, la suma total de los pesos, una lista para los comentarios y un diccionario para hacer el desglose de la calificación total en sus calificaciones individuales.

A continuación, se emplea un bucle for análogo al usado en las funciones de evaluación, que recorre las calificaciones de cada parámetro y sus insights asociados, y que asocia una puntuación de 1 a 4 a cada calificación. Estos puntos se multiplican por el peso establecido, que se extrae del diccionario correspondiente según el nombre del parámetro. Puntuación y pesos se suman de manera iterativa y se almacenan en las variables declaradas. Finalmente, se estructuran los insights asociados a cada calificación ('<Parámetro>: <calificación>-<comentario>') y se arma el diccionario de calificaciones individuales para el desglose.

El sistema de calificación a partir de la puntuación obtenida es análogo al de las funciones de evaluación de parámetros: En primer lugar calculamos la puntuación total como la suma de los puntos ponderados entre la suma de todos los pesos, y compara en una estructura condicional con los límites definidos con el algoritmo, asociando un nivel de calificación.

Pero, tenemos que tener en cuenta que los límites de calificación del rendimiento dependen de la calificación del VO<sub>2</sub> máximo del usuario, por lo que en la función *calificar\_rendimiento* se incluye un bloque condicional que adapta estos límites, definidos como diccionario.

- **Resultados.** Una vez contamos con la calificación global, se arma el diccionario *resultados\_dia*, que apila por día de la semana la calificación global, el desglose en calificaciones individuales y todos los insights correspondientes a dicho día con un salto de línea.

Los resultados diarios de todos los días se unen a la lista *resultados*, que se devuelve convertida en DataFrame. Para la función de recuperación, se añade un paso extra en el que nos aseguramos de que no se incluyen las columnas de estrés y BB si no son válidas, devolviendo únicamente las columnas que no contienen los nombres de estos parámetros.

Finalmente, estos resultados se almacenan en una variable llamando a la función.

Una vez conocemos su estructura, se muestra el código de la función *calificar\_recuperacion*. La función *calificar\_rendimiento* se adjunta en el apéndice C:

Listing 4.13: Función de calificación de la recuperación

```
#Funcion para evaluar la recuperacion.  
def calificar_recuperacion(resultados_sueno, resultados_hrv,  
    resultados_estres_bb, pesos_descanso, pesos_entrenamiento_post_descanso,  
    pesos_entrenamiento_post_entrenamiento, valid_stress):
```

```

resultados = []

if not valid_stress: #Indicamos al usuario si sus datos de estres no son
    validos
    print('Datos de Estres y Body Battery no disponibles')

for index in range(len(resultados_fc)):
    dia = resultados_fc.loc[index, 'Dia']

    ## Distincion por tipo de dia para extraccion de datos ##

    if dia in ["Miercoles", "Sabado"]:
        #Dias de descanso
        parametros = {
            'Calidad	del	Sueno': (resultados_sueno.loc[index, 'Calificacion_Global'], resultados_sueno.loc[index, 'Insight_Sueno']),
            'HRV': (resultados_hrv.loc[index, 'Calificacion_HRV'], resultados_hrv.loc[index, 'Insight_HRV']),
            'Body_Battery_Inicial': (resultados_estres_bb.loc[index, 'Calificacion_Body_Battery_Max'], resultados_estres_bb.loc[index, 'Insight_Body_Battery_Max'])
        }

        if valid_stress: #Anadimos estres y BB solo si esta disponible
            parametros['Body_Battery_Final'] = (resultados_estres_bb.loc[
                index, 'Calificacion_Body_Battery_Final'],
                resultados_estres_bb.loc[index, 'Insight_Body_Battery_Final'])

            parametros['Estres_Dia_Descanso'] = (resultados_estres_bb.loc[
                index, 'Calificacion_Estres_Medio_Dia_Descanso'],
                resultados_estres_bb.loc[index, 'Insight_Estres_Medio_Dia_Descanso'])

        pesos_parametros = pesos_descanso

    else:
        #Dias de entrenamiento
        if index > 0 and resultados_sueno.loc[index - 1, 'Dia'] in ["Miercoles", "Sabado"]:
            #Despues de dia de descanso
            parametros = {
                'Calificacion_Recuperacion_Dia_Descanso_Anterior': (
                    resultados[index - 1]['Calificacion_Recuperacion'], None),
                'Calidad	del	Sueno': (resultados_sueno.loc[index, 'Calificacion_Global'], resultados_sueno.loc[index, 'Insight_Sueno']),
                'HRV': (resultados_hrv.loc[index, 'Calificacion_HRV'], resultados_hrv.loc[index, 'Insight_HRV']),
                'Body_Battery_Inicial': (resultados_estres_bb.loc[index, 'Calificacion_Body_Battery_Max'], resultados_estres_bb.loc[index, 'Insight_Body_Battery_Max'])
            }

```

```

    }

    if valid_stress: #Anadimos estres y BB solo si esta disponible
        parametros['Body_Battery_Pre-Entrenamiento'] = (
            resultados_estres_bb.loc[index, 'Calificacion_Body_Battery_Pre-Entrenamiento'],
            resultados_estres_bb.loc[index, 'Insight_Body_Battery_Pre-Entrenamiento'])
        parametros['Estres_Pre-Entrenamiento'] = (
            resultados_estres_bb.loc[index, 'Calificacion_Estres_Pre-Entrenamiento'],
            resultados_estres_bb.loc[index, 'Insight_Estres_Pre-Entrenamiento'])

    pesos_parametros = pesos_entrenamiento_post_descanso

else: #Despues de dia de entrenamiento
    parametros = {
        'Calidad	del_Sueno': (resultados_sueno.loc[index, 'Calificacion_Global'],
                               resultados_sueno.loc[index, 'Insight_Sueno']),
        'HRV': (resultados_hrv.loc[index, 'Calificacion_HRV'],
                 resultados_hrv.loc[index, 'Insight_HRV']),
        'Body_Battery_Inicial': (resultados_estres_bb.loc[index, 'Calificacion_Body_Battery_Max'],
                                  resultados_estres_bb.loc[index, 'Insight_Body_Battery_Max'])
    }

    if valid_stress: #Anadimos estres y BB solo si esta disponible
        parametros['Body_Battery_Final_Dia_Anterior'] = (
            resultados_estres_bb.loc[index, 'Calificacion_Body_Battery_Final_Dia_Anterior'],
            resultados_estres_bb.loc[index, 'Insight_Body_Battery_Final'])
        parametros['Body_Battery_Pre-Entrenamiento'] = (
            resultados_estres_bb.loc[index, 'Calificacion_Body_Battery_Pre-Entrenamiento'],
            resultados_estres_bb.loc[index, 'Insight_Body_Battery_Pre-Entrenamiento'])
        parametros['Estres_Post_Entrenamiento_Dia_Anterior'] = (
            resultados_estres_bb.loc[index, 'Calificacion_Estres_Post_Entrenamiento_Dia_Anterior'],
            resultados_estres_bb.loc[index, 'Insight_Estres_Post_Entrenamiento_Dia_Anterior'])
        parametros['Estres_Pre-Entrenamiento'] = (
            resultados_estres_bb.loc[index, 'Calificacion_Estres_Pre-Entrenamiento'],
            resultados_estres_bb.loc[index, 'Insight_Estres_Pre-Entrenamiento'])

    pesos_parametros = pesos_entrenamiento_post_entrenamiento

## Sistema de Calificacion ##

#Calculo de la puntuacion
puntuacion_total = 0
total_pesos = 0
insights = []
calificaciones_individuales = {} #Inicializamos el diccionario para

```

```

desglose de la calificacion global

for param, (calificacion, comentario) in parametros.items(): #
    Recorremos las calificaciones y los comentarios para puntuar
    if pd.notna(calificacion):
        if calificacion == 'Excelente':
            puntos = 4
        elif calificacion in ['Bueno', 'Buena']:
            puntos = 3
        elif calificacion == 'Regular':
            puntos = 2
        else:
            puntos = 1

        #Se pondra segun el peso del parametro
        peso = pesos_parametros.get(param, 1) # Si no hay, por defecto
        #coge 1
        puntuacion_total += puntos * peso
        total_pesos += peso

        #Estructura final de los insights
        if comentario:
            insights.append(f'{param}:{calificacion}-{comentario}')

        #Se anade la calificacion a un diccionario para el desglose de
        #la calificacion global
        calificaciones_individuales[f'Calificacion_{param}'] =
            calificacion

#Calificacion segun la puntuacion

puntuacion_promedio = puntuacion_total / total_pesos
if puntuacion_promedio >= 3.5:
    calificacion_global = 'Excelente'
elif puntuacion_promedio >= 2.5:
    calificacion_global = 'Buena'
elif puntuacion_promedio >= 1.5:
    calificacion_global = 'Regular'
else:
    calificacion_global = 'Mala'

## Resultados ##

#Anadimos al diccionario
resultado_dia = {
    'Dia': dia,
    'Calificacion_Recuperacion': calificacion_global,
    **calificaciones_individuales, # Anadimos el desglose en
        # calificaciones individuales
    'Insight_Recuperacion': "\n".join(insights)
}

```

```

    }

    resultados.append(resultado_dia)

    #Convertir lista de resultados en DataFrame
    df_resultados = pd.DataFrame(resultados)

    #Nos aseguramos de que se filtran columnas de estres y BB si el estres no
    #es valido
    if not valid_stress:
        df_resultados = df_resultados.loc[:, ~df_resultados.columns.str.
            contains('Estres|Body\Battery\Pre|Body\Battery\Final')]

    return df_resultados

#Se almacena el DF con los resultados llamando a la funcion
resultados_recuperacion = calificar_recuperacion(
    resultados_sueno, resultados_hrv, resultados_estres_bb,
    pesos_descanso, pesos_entrenamiento_post_descanso,
    pesos_entrenamiento_post_entrenamiento,
    estres_valido
)

```

Listing 4.13: Función de calificación de la recuperación

## Visualización de los resultados

La última parte de la implementación consiste en la visualización de los resultados obtenidos, de tal manera que le resulte intuitiva y simple al usuario, pero que al mismo tiempo permita a un especialista aconsejarle sobre los aspectos que puede mejorar mediante el sistema de comentarios.

En primer lugar, se ha construido una tabla que concatena la información de la recuperación y el rendimiento en el entrenamiento por día de la semana, para que ambas variables puedan relacionarse. Para ello, se usa la función *merge* para aunar los DataFrames resultantes de las funciones de calificación por día por la columna *Día*. Sustituimos con *fillna* los nulos por guiones para que no aparezcan valores 'NN' o 'None' en la tabla.

Se emplea un código de colores asociado al nivel de calificación para facilitar la lectura de la tabla. Para ello, se usa la función *color\_cell*, que asocia un color de fondo de celda a la calificación con una estructura condicional. Además, se ha añadido una función *estilo\_divisor*, para marcar los bordes laterales de una celda, con objetivo de remarcar las calificaciones globales de rendimiento y recuperación.

Para aplicar el estilo (colores y bordes) a la tabla, se usa la función *applymap*, en la que se especifica como parámetro de entrada la función de estilizado y la/s celda/s a las que se le aplica (en el parámetro *subset*).

Por otro lado, los comentarios se muestran con un sistema de desplegables por día, diferenciando entre recuperación y rendimiento. Para ello, se usan los siguientes paquetes en un bucle que recorre los días:

- *ipywidgets*, para el uso de elementos interactivos en Python como botones o, en este caso, un 'acordeón'. El acordeón es un panel de texto expansible que, cuando en el tamaño

ajustado no cabe el texto, despliega una barra lateral deslizable para recorrer todo su contenido [63].

En el parámetro *children* se indican los paneles que se mostrarán. En este caso son de tipo *Text Area*, donde se indica qué va a imprimirse, el tamaño del layout, si se habilita o deshabilita el modo edición y el título.

- Función *display* del módulo *IPython.display*. Permite mostrar objetos, contenido multimedia y HTML. Lo usaremos con la tabla y los desplegables.

A parte, se imprime el valor del VO<sub>2</sub> máximo del usuario y el mensaje de no disponibilidad de los datos de estrés.

El código para la implementación es el siguiente:

Listing 4.14: Visualización de la tabla de calificaciones y los desplegables de insights

```
import ipywidgets as widgets
from IPython.display import display

## Confeccion de la tabla ##

#Funcion para asignar un color de celda a cada calificacion
def color_cell(calificacion):
    if calificacion == 'Mala' or calificacion == 'Malo':
        return 'background-color: red'
    elif calificacion == 'Regular':
        return 'background-color: yellow'
    elif calificacion == 'Buena' or calificacion == 'Bueno':
        return 'background-color: lightgreen'
    elif calificacion == 'Excelente':
        return 'background-color: green'
    else:
        return ''

#Funcion para anadir lineas a las celdas que indiquemos
def estilo_divisor(val):
    return 'border-left: 2px solid black; border-right: 2px solid black;'

#Juntamos ambos DFs en uno, sustituimos los nulos por guion
resultados_completos = pd.merge(resultados_recuperacion,
    resultados_rendimiento, on = 'Dia', how='outer').fillna('-')

#Excluimos los insights de la grafica
tabla_sin_insights = resultados_completos.drop(columns=[ 'Insight_Recuperacion',
    , 'Insight_Rendimiento'])

#Aplicamos el estilo de color y los bordes a las celdas indicadas
styled_tabla_sin_insights = tabla_sin_insights.style.applymap(
    color_cell, subset=tabla_sin_insights.columns.difference(['Dia']))
).applymap(
    estilo_divisor, subset=[ 'Calificacion_Recuperacion'])
).applymap(
    estilo_divisor, subset=[ 'Calificacion_Rendimiento'])
```

```
)  
  
#Mostramos la tabla con el estilo aplicado  
display(styled_tabla_sin_insights)  
  
## Sistema de desplegables ##  
for idx, row in resultados_completos.iterrows():  
    dia = row['Dia']  
    print(f"Detalles del {dia}:")  
  
    #Desplegable para "Insight Recuperacion"  
    if pd.notna(row['Insight_Recuperacion']):  
        desplegable_recuperacion = widgets.Accordion(children=[widgets.Textarea  
            (value=row['Insight_Recuperacion'], layout=widgets.Layout(width='500  
            px', height='100px'), disabled=True)])  
        desplegable_recuperacion.set_title(0, f'Insights_Recuperacion-{dia}')  
        display(desplegable_recuperacion)  
  
    #Desplegable para "Insight Rendimiento"  
    if pd.notna(row['Insight_Rendimiento']):  
        desplegable_rendimiento = widgets.Accordion(children=[widgets.Textarea(  
            value=row['Insight_Rendimiento'], layout=widgets.Layout(width='500px  
            ', height='100px'), disabled=True)])  
        desplegable_rendimiento.set_title(0, f'Insights_Rendimiento-{dia}')  
        display(desplegable_rendimiento)  
  
    print("\n" + "-"*100 + "\n") #Anadimos guiones para separar entre dias
```

Listing 4.14: Visualización de la tabla de calificaciones y los desplegables de insights



# 5. Análisis de Resultados

En este capítulo se exponen los resultados obtenidos para cada uno de los sujetos participantes en el estudio, a partir de los cuales un experto en ciencias del deporte o la salud podría obtener una valoración del rendimiento deportivo y la recuperación del atleta para tomar las decisiones sobre su plan de entrenamiento.

El procedimiento de análisis consiste en tres puntos principales:

- **Análisis del rendimiento y la recuperación en sus parámetros individuales.** Para cada usuario, se hará un análisis de la evolución diaria y semanal de los parámetros estudiados, mostrando gráficamente la evolución diaria y semanal de los mismos.  
Como se comentará en el Capítulo 6, **el análisis llevado a cabo no es de rigor y debería dejarse a cargo de especialistas.**

Simplemente, se comentará la tendencia de crecimiento de las variables, donde aparecen sus máximos y mínimos y cómo se comparan con los valores de referencia, para hacer una relación con los resultados de los estudios de la literatura revisada.

- **Análisis global. Resultados del algoritmo propuesto.** El análisis global comprende la comprobación de los resultados de la implementación del algoritmo propuesto (descrito en la sección 3.3) en Python, el proceso de análisis conjunto de todos los parámetros estudiados mediante el sistema de calificación del algoritmo, y el análisis cualitativo de la eficiencia del mismo analizando la correlación entre los resultados y la experiencia registrada de los usuarios.
- **Análisis del rendimiento de la red personal.** Además del análisis deportivo, se cuestionará si la red personal utilizada durante el estudio cumple con los principios fundamentales de los dispositivo wearable, expuestos en la sección 2.1.1, mediante el análisis de la eficiencia de su autonomía y los resultados de la encuesta SUS planteada en la sección 3.4.

Se recuerda que los datos de los usuarios están tratados de forma anónima para proteger su privacidad de acuerdo a la LOPDGDD.

## 5.1. Análisis del rendimiento y la recuperación en sus parámetros individuales.

### 5.1.1. Variables Basales

#### Frecuencia Cardíaca y HRV Basal

Este subapartado está dedicado a los procedimientos de obtención y cálculo de los parámetros de frecuencia cardíaca necesarios para el estudio.

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

En primer lugar, para obtener el valor de su **frecuencia cardíaca en reposo**, los usuarios se pusieron el reloj durante una semana. Mediante la información del sensor OHR, Garmin devuelve el valor medio de este parámetro durante dicho periodo de tiempo.

Por otro lado, calculamos el rango de variabilidad de la frecuencia cardíaca en reposo haciendo 5 mediciones a lo largo de un día en el que el usuario no realiza actividad ni la hizo el día previo. Nos quedamos con el valor máximo y el mínimo de esas mediciones como límites del umbral de reposo.

En la siguiente tabla se recogen los resultados:

Usuario	FC Basal (ppm)	Rango HRV (ms)
Usuario 1	52.5	53–85
Usuario 2	49	61–92

Cuadro 5.1: FC y HRV basal por usuario

### VO<sub>2</sub> Máximo Inicial

Tras la prueba de carrera, el usuario 1 registró un VO<sub>2</sub> máximo relativo de 43 ml/kg/min, un valor que Garmin califica como 'Aceptable' y que se haya por debajo del medio obtenido en los estudios de la literatura revisada [15, 16, 17, 18, 19, 20, 21], de entre  $54.5 \pm 4.5$  y  $63.8 \pm 4.7$  ml/min/kg.

El usuario 2 registró un VO<sub>2</sub> máximo inicial de 46 ml/kg/min, calificado como 'Bueno', aunque también por debajo del recomendado por los estudios de boxeo.

Usuario	VO <sub>2</sub> Máximo registrado (ml/kg/min)	VO <sub>2</sub> Máximo Ref. (ml/kg/min)
Usuario 1	43	54.5–63.8
Usuario 2	46	

Cuadro 5.2: Comparación del VO<sub>2</sub> máximo inicial de los usuarios con el de referencia

#### 5.1.2. Frecuencia Cardíaca

En este apartado analizamos los resultados de frecuencia cardíaca obtenidos en los entrenamientos de boxeo.

##### FC media, máxima y mínima

En primer lugar, se muestran las gráficas de frecuencia cardíaca monitorizada por segundo durante el entrenamiento, destacando los valores máximo y mínimo, así como el valor medio de toda la sesión.

Podemos comprobar el impacto en la gráfica de la estructura temporal interválica de los entrenamientos de deporte de contacto, obteniendo una representación de la frecuencia cardíaca con forma de 'peine', diferenciando con claridad los períodos de actividad y de descanso.

La pausa después del calentamiento, con la finalidad del equipamiento del vendaje y los guantes de boxeo, se refleja en una bajada de pulsaciones de unos 7 minutos (2 asaltos) tras los 7 asaltos de calentamiento.

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

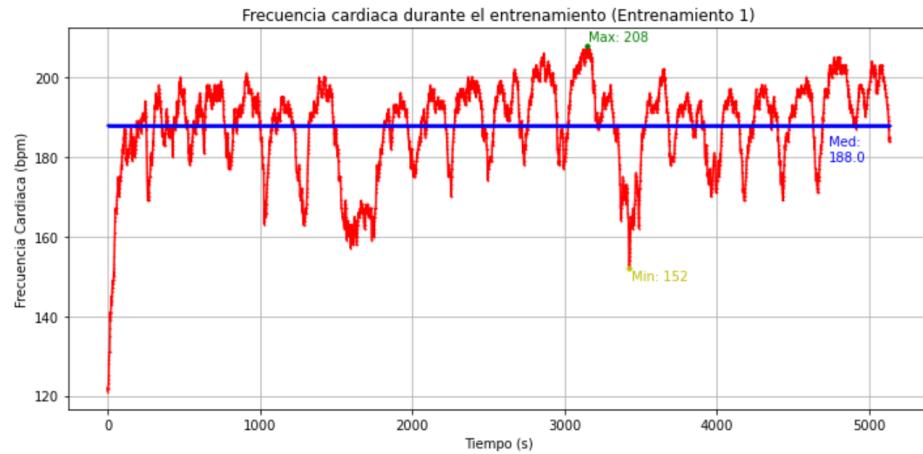


Figura 5.1: Frecuencia cardíaca durante el entrenamiento 1, usuario 1

En la siguiente gráfica se registra la evolución de los valores medios, máximos y mínimos de la frecuencia cardíaca durante la semana de entrenamiento, para poder analizar el efecto negativo de la fatiga acumulada, así como el positivo generado por la mayor recuperación tras el día de descanso (entre los entrenamientos 2 y 3).

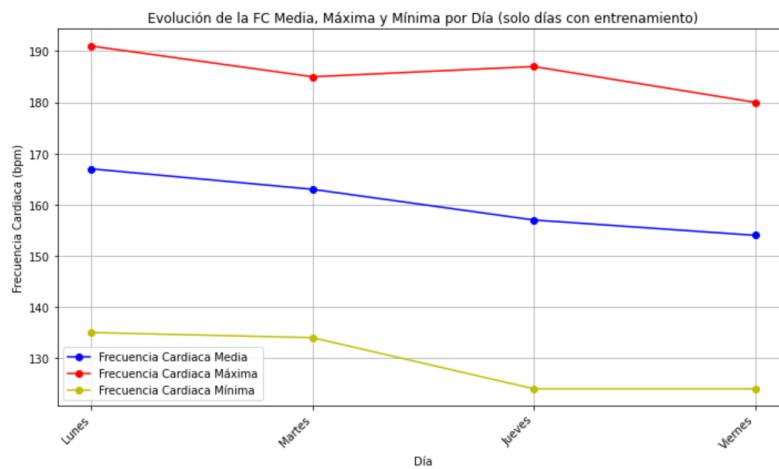


Figura 5.2: Evolución semanal de la frecuencia cardíaca media, máxima y mínima, usuario 1

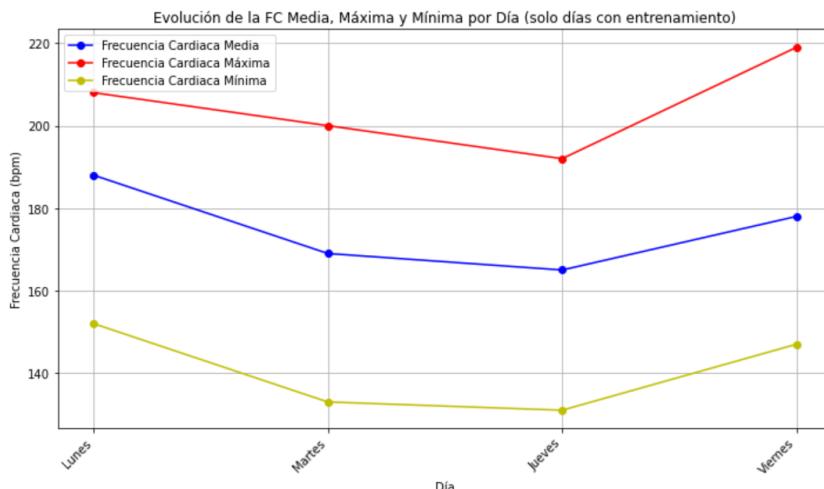


Figura 5.3: Evolución semanal de la frecuencia cardíaca media, máxima y mínima, usuario 2

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

Los valores para ambos usuarios se recogen en las siguientes tablas. Se marca el valor máximo de FC registrado para el posterior cálculo de la HRR y las zonas de FC óptimas.

Entrenamiento	FC media (bpm)	FC máx (bpm)	FC mín (bpm)
E1	167	<b>191</b>	135
E2	163	185	134
E3	167	187	124
E4	154	180	124

Cuadro 5.3: FC media, máxima y mínima registrada en los entrenamientos, usuario 1

Entrenamiento	FC media (bpm)	FC máx (bpm)	FC mín (bpm)
E1	188	208	152
E2	169	200	133
E3	165	192	131
E4	178	<b>219</b>	147

Cuadro 5.4: FC media, máxima y mínima registrada en los entrenamientos, usuario 2

A partir de estos datos podemos concluir lo siguiente:

- La mayoría de valores máximos durante el entrenamiento fueron registrados en los últimos asaltos del mismo, cuando la fatiga acumulada es mayor. Además se realizan los ejercicios de mayor intensidad (potencia interválica y sparring). El usuario 2 alcanza el valor máximo de la semana durante un asalto de sparring el viernes.
- Podemos comprobar que los valores de las tres variables de FC del primer día (día de entrenamiento tras descanso) son las máximas registradas para el usuario 1. El usuario 2 registra el máximo de FC media. Ambos usuarios se encontraban con energía antes de comenzar dicho entrenamiento, por lo que entrenaron a gran intensidad.
- Puede observarse que tras el día de descanso del miércoles (E3) se producen cambios más significativos respecto a otros días. La frecuencia cardíaca media y mínima decaen en el caso del usuario 1. El usuario 2 registra los valores más bajos (marcados en verde) de los 3 parámetros ese día. Ambos usuarios comentaron que se sentían descansados. Por otro lado, el valor máximo alcanzado por el usuario 1 es mayor. Ese día también entrenó a gran intensidad.
- En general, la tendencia de los tres parámetros es descendiente, salvo para el día final del usuario 2, donde alcanzó valores muy altos durante el sparring.

Como análisis general, y sin menoscabo que sea validado por un especialista y en un análisis más detallado, destaca el posible impacto de los días de descanso y de la fatiga acumulada. La fatiga también tiene efecto en la intensidad con la que los usuarios afrontaban el entrenamiento, posible causa de la tendencia descendiente. La intensidad requerida en los ejercicios finales también se ha apreciado en los resultados.

### HRR y zonas de FC óptima

En este apartado se calculan las zonas de FC óptima a partir de la HRR y se comparan con los valores de FC media registrados a lo largo de las sesiones de boxeo (5.3 y 5.4).

Para calcular la HRR y las zonas de FC óptima se sigue el procedimiento de Karvonen descrito en la sección 2.2.3. En la siguiente tabla se recogen los resultados para ambos sujetos:

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

Usuario	FC basal (bpm)	FC máx (bpm)	HRR (bpm)	Zona FC óptima (bpm)
Usuario 1	52.5	194.7 (Tanaka)	142.2	152.04–173.37
Usuario 2	49	219 (Exp)	170	171.5–197

Cuadro 5.5: HRR y zonas de FC óptima por usuario

Finalmente, analizamos si la frecuencia cardíaca media de los entrenamientos se encuentra dentro de las zonas de FC óptima.

Usuario	Frecuencia Cardíaca Media(bpm)				Zona FC óptima (bpm)
	E1	E2	E3	E4	
Usuario 1	167	163	167	154	152.04–173.37
Usuario 2	188	169	165	178	171.5–197

Cuadro 5.6: Comparativa de FC media experimental con zonas de FC óptima

Podemos comprobar que el usuario 2 no entrena a su frecuencia cardíaca óptima durante los entrenamientos 2 y 3. Esto sugiere que la intensidad no fue la adecuada.

### FC media por tipo de asalto

A continuación, se muestran los resultados del análisis de FC por tipo de asalto, tras la implementación de la detección y procesamiento de picos y valles. Como ejemplo, se muestra la gráfica obtenida para el entrenamiento del lunes del usuario 1:

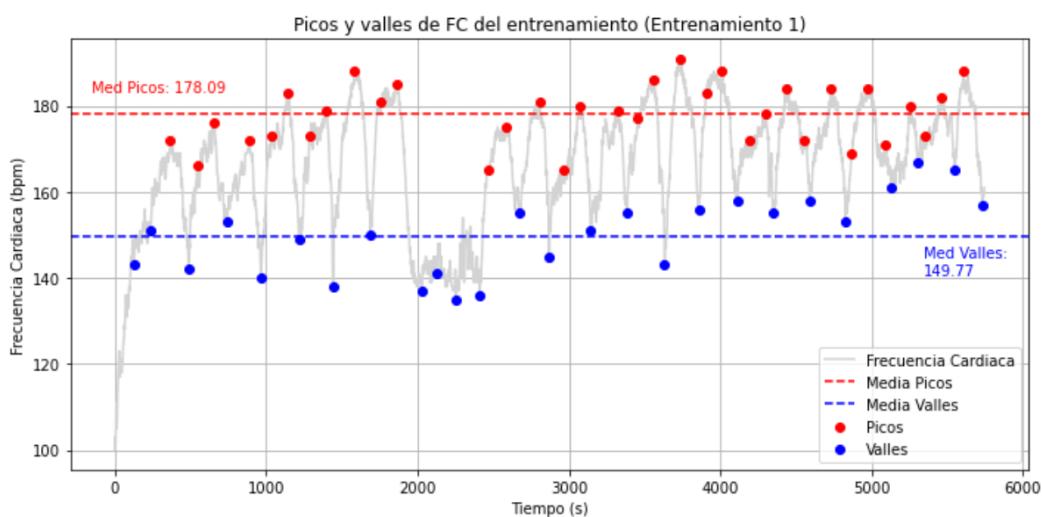


Figura 5.4: Picos y valles de frecuencia cardíaca durante el entrenamiento 1, usuario 1

En las siguientes tablas se muestran los valores medios alcanzados por los usuarios en cada entrenamiento, así como los valores de referencia de los estudios que realizan esta separación por intervalos ([25] para usuario 1, [26] para usuario 2).

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

Tipo de Intervalo	Frecuencia Cardíaca (bpm)				Valor Medio	Valor de Ref. [25]
	E1	E2	E3	E4		
Actividad	178.10	174.21	169.63	164.40	171.59	171
Recuperación	150.20	148.03	137.0	136.75	143.99	161.5

Cuadro 5.7: Comparativa teórica/experimental de valores medios de FC por tipo de intervalo, usuario 1

Tipo de Intervalo	Frecuencia Cardíaca (bpm)				Valor Medio	Valor de Ref. [26]
	E1	E2	E3	E4		
Actividad	197.19	181.48	177.65	191.44	186.78	181.3
Recuperación	172.13	148.15	143.05	161.08	156.10	NE

Cuadro 5.8: Comparativa teórica/experimental de valores medios de FC por tipo de intervalo, usuario 2

El usuario 1 alcanza unas pulsaciones menores que las de referencia. Las pulsaciones medias de sus asaltos de intensidad están en el rango de los estudios de referencia de su nivel. El usuario presenta una bajada significativa en los bpm de ambos asaltos tras el día de descanso.

El usuario 2 supera ligeramente las pulsaciones medias de los asaltos de actividad de boxeadores de su nivel, resultado influenciado por los altos valores alcanzados el día de sparring. El usuario alcanza los valores mínimos de FC tras el día de descanso.

Si representamos gráficamente esta evolución semanal, podemos apreciar el impacto positivo del día de descanso en el desempeño de ambos usuarios:

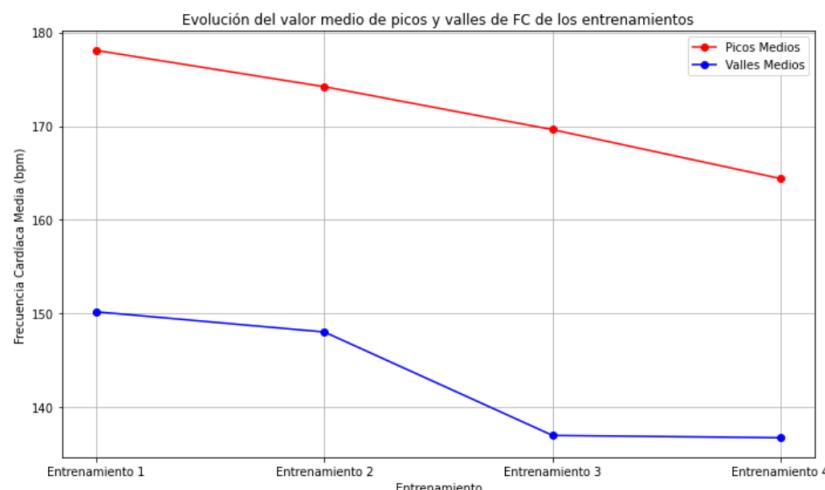


Figura 5.5: Evolución semanal de la media de picos y valles de frecuencia cardíaca, usuario 1

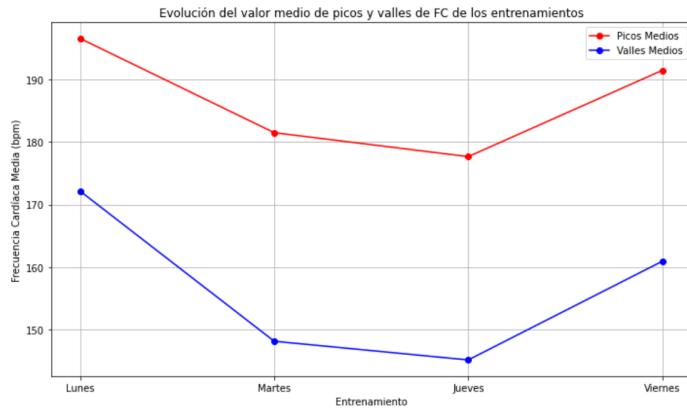


Figura 5.6: Evolución semanal de la media de picos y valles de frecuencia cardíaca, usuario 2

### 5.1.3. Variabilidad de la Frecuencia Cardíaca

En este apartado se muestran los resultados de los análisis de HRV registrados en las pruebas de Resumen de Salud.

A continuación se muestra la evolución de dicho parámetro un día de entrenamiento, 5.7, y uno de descanso, 5.8, pudiendo comprobar que la implementación de la distinción en el número de valores según el tipo de día ha sido correcta:

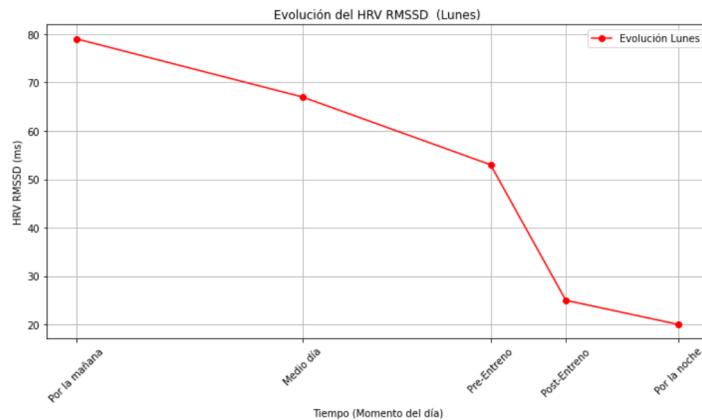


Figura 5.7: Evolución del HRV un día de entrenamiento, usuario 1

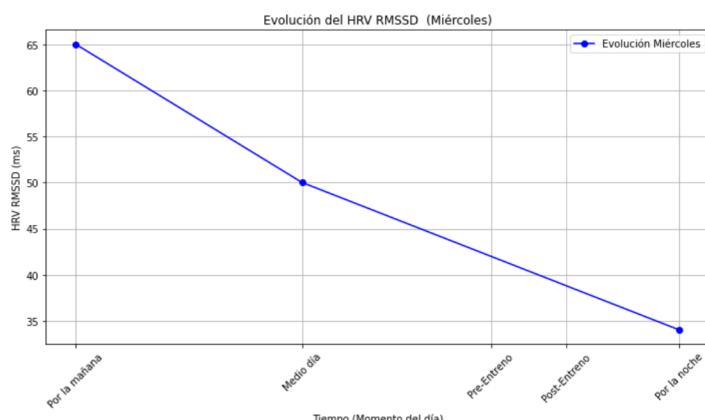


Figura 5.8: Evolución del HRV un día de descanso, usuario 1

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

Podemos observar que el máximo de este valor se alcanza nada más se despierta el usuario, cuando se encuentra más recuperado. Este valor disminuye a lo largo del día.

Por otro lado, comprobamos que el entrenamiento tiene un impacto muy significativo en la variabilidad de la frecuencia cardíaca, alcanzándose los valores diarios mínimos. Este valor tiende a estabilizarse pasadas unas horas, aunque a un valor menor que el que se registra antes de entrenar.

La comparativa de estos valores a lo largo de la semana de entrenamientos se muestra en las figuras 5.9 y 5.10, y se recoge en las tablas 5.9 y 5.10:

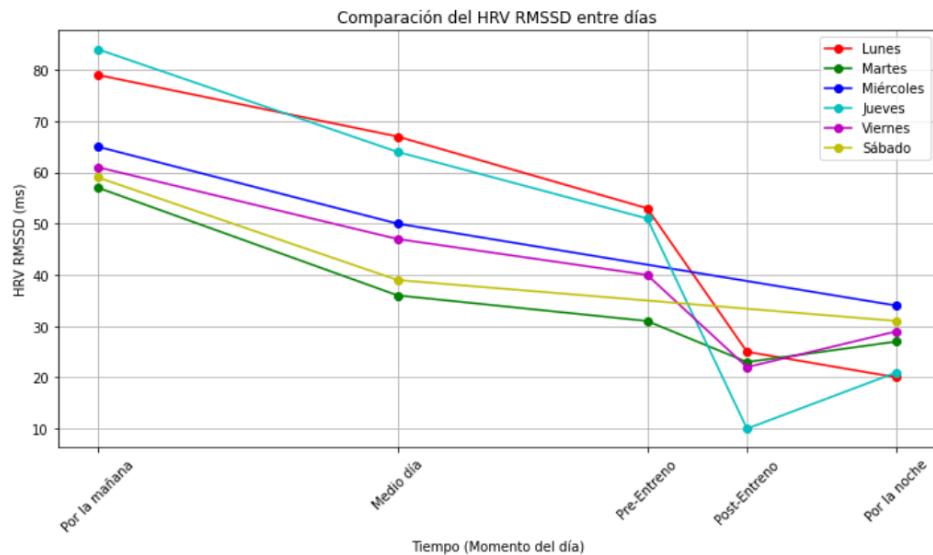


Figura 5.9: Evolución temporal del HRV durante la semana de entrenamiento, usuario 1

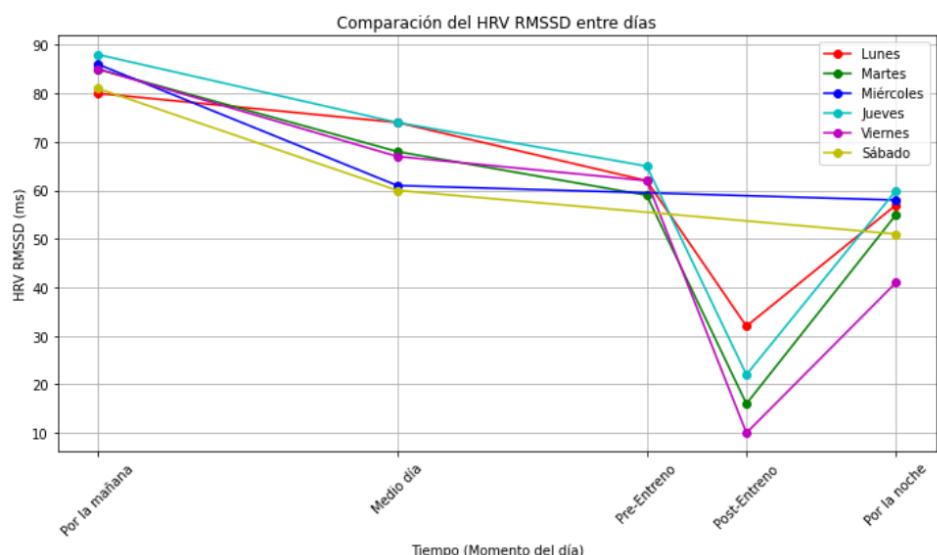


Figura 5.10: Evolución temporal del HRV durante la semana de entrenamiento, usuario 2

Día	Variabilidad de la frecuencia cardíaca (ms)				
	Al levantarse	Medio día	Pre-entreno	Post-entreno	Al acostarse
Lunes	79	67	53	25	20
Martes	57	36	31	23	27
Miércoles	65	50	—	—	34
Jueves	84	64	51	10	21
Viernes	61	47	40	22	29
Sábado	59	39	—	—	31

Cuadro 5.9: Evolución temporal del HRV durante la semana de entrenamiento, usuario 1

Día	Variabilidad de la frecuencia cardíaca (ms)				
	Al levantarse	Medio día	Pre-entreno	Post-entreno	Al acostarse
Lunes	80	74	62	32	57
Martes	85	68	59	16	55
Miércoles	86	61	—	—	58
Jueves	88	74	65	22	60
Viernes	81	60	51	10	42
Sábado	81	60	—	—	41

Cuadro 5.10: Evolución temporal del HRV durante la semana de entrenamiento, usuario 2

Estos resultados nos sugieren lo siguiente:

1. El impacto de la fatiga se refleja en mucha mayor medida en el HRV del usuario 1, que alcanza con mayor facilidad valores inferiores a su rango basal. Esto sugiere que el nivel de entrenamiento (o la experiencia) puede suponer un factor en la recuperación del organismo.
2. Los valores máximos de HRV se alcanzan cuando se ha descansado el día anterior, posible síntoma de la recuperación del organismo.  
Los valores mínimos se encuentran en los días de mayor fatiga acumulada (días de entrenamiento tras entrenamiento).
3. Los días tras descanso el balance de HRV pre y post entrenamiento es también mayor.
4. Los días de descanso el balance diario de HRV es mucho menor que los de entrenamiento, ya que en este periodo se produce el mayor gasto.

Volvemos a presenciar en los datos registrados que el descanso entre sesiones y la fatiga acumulada tienen efecto en el funcionamiento del organismo, en este caso sobre la energía corporal. Este funcionamiento podría ser explicado en mayor detalle por un especialista.

#### 5.1.4. Batería Corporal

Esta sección recoge los resultados de la monitorización de la batería corporal a lo largo de los días de la semana de estudio.

La relación de este parámetro con el nivel de estrés han llevado a la representación conjunta de ambos parámetros. Las gráficas mostradas en esta sección serán conjuntas a las que usaremos posteriormente en el apartado dedicado al análisis de estrés diario, 5.1.7.

Los resultados de monitorización de este parámetro (y del estrés diario) para el primer usuario son negativos, en referencia a que **los datos obtenidos están corruptos**. Como se muestra en la figura 5.11 y para todos los días de la semana, el registro de datos de estos

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

parámetros se cortó antes de tiempo. Como veremos para el segundo usuario, este parámetro se mide de manera diaria de 00:00h a 23:59h, pero para el usuario 1 solo obtenemos datos completos para el jueves y el sábado.



Figura 5.11: Datos corruptos de estrés diario y batería corporal, usuario 1

Por tanto, nos limitaremos a analizar este parámetro (y el estrés diario) para el segundo usuario. Uno de los archivos presenta el mismo error que los del primer usuario, pero corresponde a un día sin entrenamiento, por lo que realizaremos el análisis con los datos del resto de días.



Figura 5.12: Nivel de estrés y batería corporal a lo largo del día, usuario 2

Como podemos comprobar, la batería corporal aumenta con estrés menor de 25 (azul) y disminuye con valores superiores (naranja), lo que esperábamos dado la separación de la escala en ese umbral según los comportamientos del SNA.

El máximo valor se alcanza tras el periodo de sueño y el mínimo (tras el máximo) al finalizar el día.

Analicemos la evolución semanal de los parámetros significativos, listados en la siguiente

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

tabla.

Día	BB Inicial	BB Pre entreno	BB final	Gasto BB Entrenamiento
Lunes	93	52	20	-29
Martes	86	44	11	-23
Miércoles	93	—	56	—
Jueves	99	54	26	-21
Viernes	95	51	15	-19
Sábado	85	—	54	—

Cuadro 5.11: Body battery a lo largo de la semana, usuario 2

Podemos comprobar que los días posteriores a un día de descanso el usuario cuenta con mayor reserva de energía al levantarse, antes de entrenar y al final del día. Los peores valores, en cambio, se obtienen los días con carga acumulada.

### 5.1.5. Balance Calórico

En este apartado recogemos el gasto calórico diario de toda la semana, diferenciando entre si el usuario se encontraba en reposo o en periodo de actividad y, de este, especificaremos el gasto del entrenamiento:

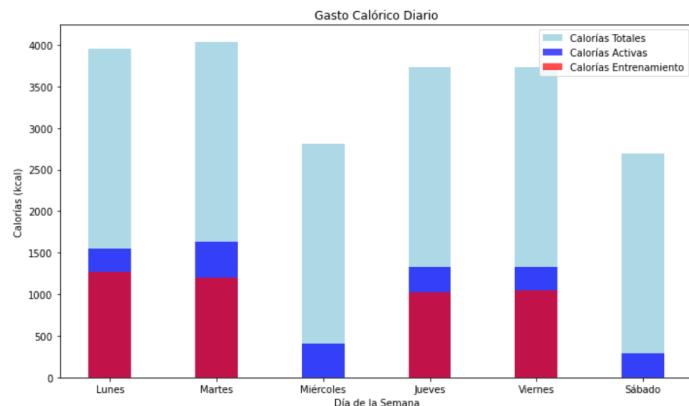


Figura 5.13: Consumo calórico diario por tipo de actividad, usuario 1

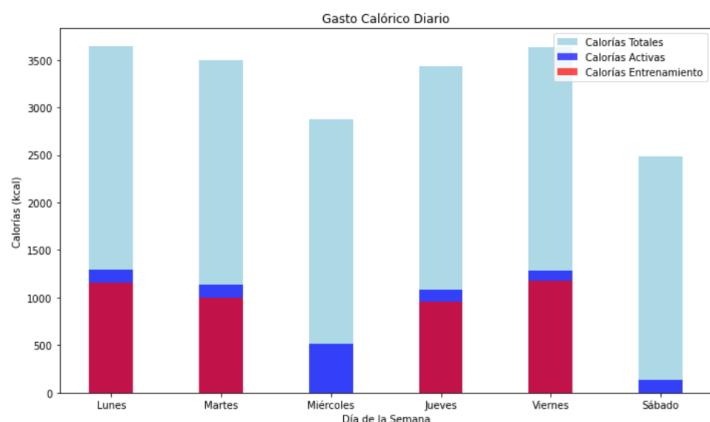


Figura 5.14: Consumo calórico diario por tipo de actividad, usuario 2

Comprobamos que incluso los días de boxeo, el mayor consumo calórico es en reposo.

### 5.1.6. Frecuencia Respiratoria

En esta sección se analiza el impacto del entrenamiento en los niveles de frecuencia respiratoria, así como el posible impacto de los días de reposo en la variación este parámetro.

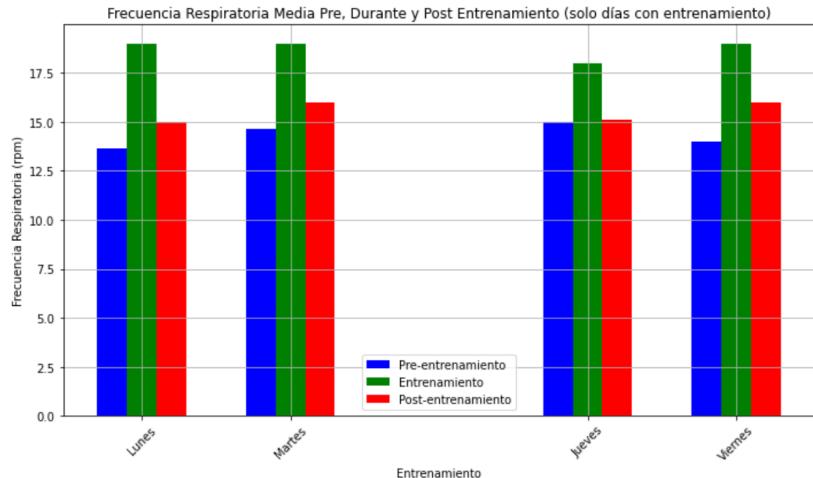


Figura 5.15: Frecuencia respiratoria media pre, durante y post entrenamiento a lo largo de la semana de estudio, usuario 1

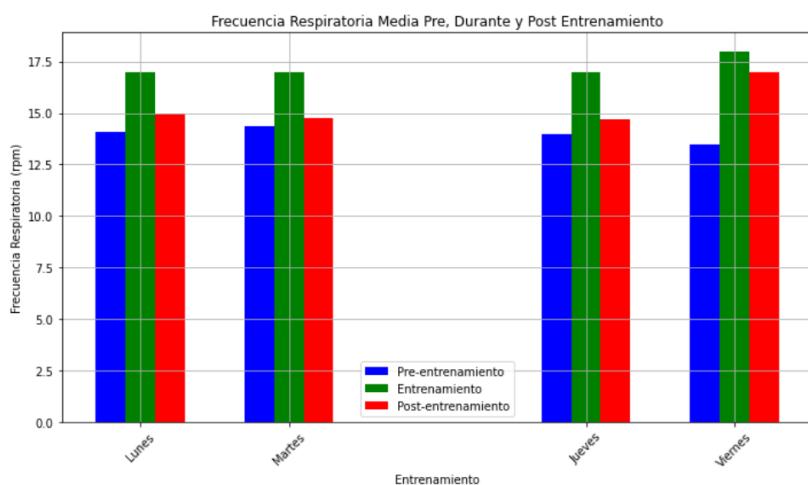


Figura 5.16: Frecuencia respiratoria media pre, durante y post entrenamiento a lo largo de la semana de estudio, usuario 2

Día (Entreno)	Frecuencia respiratoria media (rpm)			
	Pre-entreno	Entreno	Post-entreno	Balance Post-Pre
Lunes (E1)	13.65	19.00	15.00	1.35
Martes (E2)	14.60	19.00	16.00	1.40
Jueves (E3)	15.00	18.00	15.10	0.10
Viernes (E1)	14.00	19.00	16.00	2.00

Cuadro 5.12: Frecuencia respiratoria media pre, post y durante el entrenamiento, usuario 1

Día (Entreno)	Frecuencia respiratoria media (rpm)			
	Pre-entreno	Entreno	Post-entreno	Balance Post-Pre
Lunes (E1)	14.11	17.00	15.00	0.89
Martes (E2)	14.38	17.00	14.77	<b>0.39</b>
Jueves (E3)	13.99	17.00	14.72	0.73
Viernes (E1)	13.45	18.00	17.00	3.55

Cuadro 5.13: Frecuencia respiratoria media pre, post y durante el entrenamiento, usuario 2

Se puedes observar que el usuario 2, de mayor nivel de boxeo, presenta valores menores de respiración media durante el entrenamiento. Además, salvo el día de sparring, donde acabó muy fatigado por el sobre-esfuerzo, registra un menor balance entre FR pre y post entrenamiento. El día de descanso no parece influenciar en este parámetro.

Por otro lado, el usuario 1 sí que alcanza su mejor nivel de recuperación el entrenamiento posterior al día de reposo del miércoles.

Ambos usuarios presentan el peor balance pre/post entrenamiento el último día de la semana, posible efecto de la fatiga acumulada.

### 5.1.7. Nivel de Estrés

Este apartado recoge los resultados de la medición del nivel de estrés, tanto el diario como el balance pre/post entrenamiento.

#### Estrés diario

Como se ha comentado, este análisis va de la mano que el del Body Battery, por lo que referenciaremos la representación gráfica de la sección 5.1.4.

De la misma manera, los datos del usuario 1 sobre esta variable están corruptos, por lo que nos quedamos con el análisis para el segundo usuario.

En la siguiente tabla se muestra los datos de estrés diarios, diferenciando entre si se han obtenido de manera previa o posterior al entrenamiento en caso de haberlo dicho día. Los días de descanso se muestra el estrés medio diario:

Día	Estrés Pre-entreno	Estrés Post-entreno	Estrés Día Descanso
Lunes	31.68	37.18	—
Martes	28.78	42.95	—
Miércoles	—	—	34.91
Jueves	27.18	<b>25.2</b>	—
Viernes	25.9	33.28	—
Sábado	—	—	17.40

Cuadro 5.14: Nivel de estrés medio por tipo de día a lo largo de la semana, usuario 2

El usuario 2 registra el nivel de estrés post-entrenamiento más bajo una sesión tras un día de descanso.

#### Estrés pre/post entrenamiento

El nivel de estrés alcanzado en el entrenamiento lo registramos con el resumen de salud realizado justo al acabar el mismo.

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

Como podemos comprobar, se alcanza el nivel máximo de estrés medio de la escala de Garmin (99-100), incluso aunque se mide en un periodo de reposo.

Este dato no es concluyente, ya que las pulsaciones del usuario siguen siendo muy elevadas por el ejercicio. Se comentó en la sección 2.1.1 que Garmin no mide el estrés durante períodos de actividad debido a que este se confunde con el estrés.

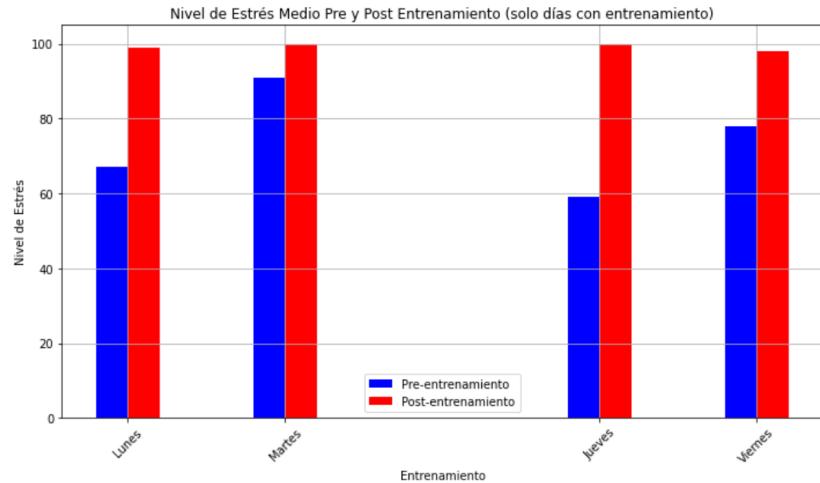


Figura 5.17: Balance de estrés pre/post entrenamiento, usuario 1

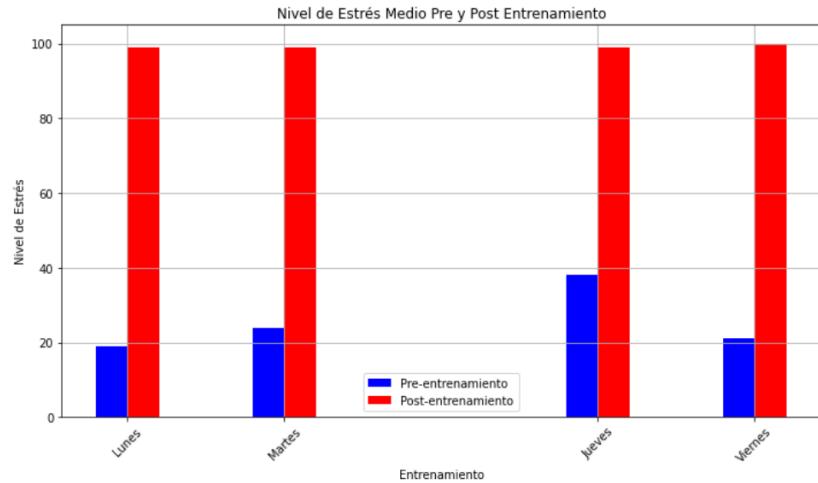


Figura 5.18: Balance de estrés pre/post entrenamiento, usuario 2

Únicamente podemos comentar la mayor relajación y menor estrés pre-entrenamiento del usuario de mayor experiencia.

### 5.1.8. Calidad del Sueño

En este apartado se califica, siguiendo la escala de puntuación de Garmin, la calidad del sueño de los usuarios a lo largo de la semana.

En primer lugar mostramos una representación gráfica de las zonas de sueño (ligero, profundo y REM) a lo largo del tiempo en un periodo de sueño. Representamos un periodo de sueño 'regular' (FAIR), 5.19 y uno excelente (Excelent), 5.20, para analizar su diferencia.

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

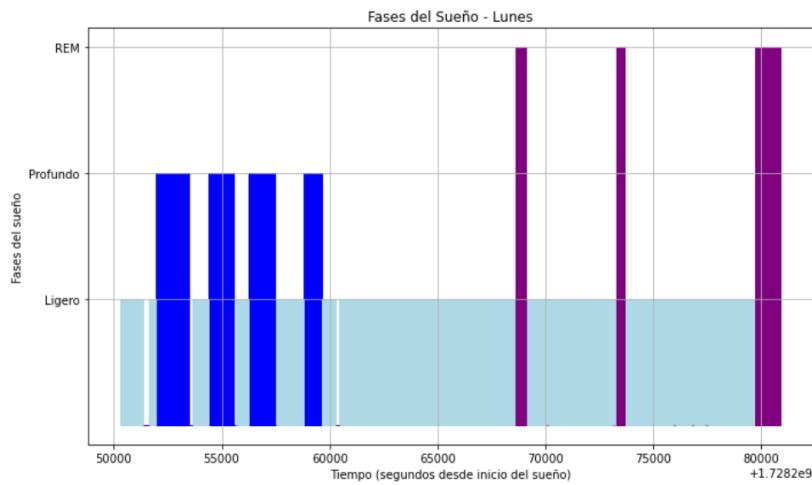


Figura 5.19: Zonas de sueño ligero, profundo y REM a lo largo de un período de sueño (FAIR), usuario 1

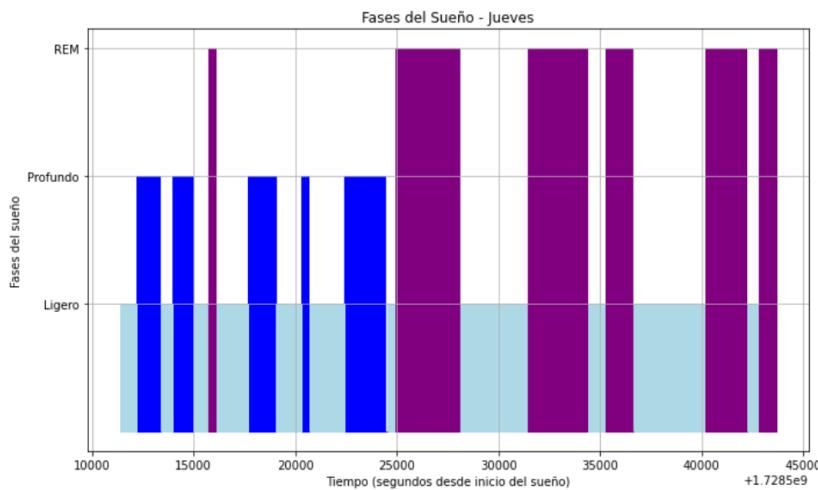


Figura 5.20: Zonas de sueño ligero, profundo y REM a lo largo de un período de sueño (EXCELENT), usuario 1

Como podemos comprobar, el porcentaje de las zonas de sueño profundo y REM tienen un gran peso en la calidad del sueño y, por tanto, en la recuperación de los atletas.

A continuación, mostramos los resultados de la monitorización de la calidad del sueño de la semana de entrenamiento completa, ya que nos brinda más parámetros para analizar.

	Día	Puntuación Global	Calificación Global	Duración Total	Estrés	Inquietud	Despertar	REM	Sueño Ligero	Sueño Profundo
0	Lunes	77	FAIR	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	POOR	FAIR	GOOD
1	Martes	70	FAIR	FAIR	EXCELLENT	FAIR	POOR	FAIR	FAIR	FAIR
2	Miércoles	65	FAIR	FAIR	EXCELLENT	POOR	POOR	FAIR	FAIR	EXCELLENT
3	Jueves	94	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	FAIR	EXCELLENT	EXCELLENT
4	Viernes	71	FAIR	FAIR	EXCELLENT	POOR	POOR	FAIR	FAIR	EXCELLENT
5	Sábado	55	POOR	FAIR	EXCELLENT	POOR	POOR	POOR	FAIR	FAIR

Figura 5.21: Calificaciones de la calidad del sueño a lo largo de la semana de entrenamientos, usuario 1

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

Día	Puntuación Global	Calificación Global	Duración Total	Estrés	Inquietud	Despertar	REM	Sueño Ligero	Sueño Profundo
0 Lunes	80	GOOD	FAIR	EXCELLENT	FAIR	FAIR	EXCELLENT	EXCELLENT	EXCELLENT
1 Martes	83	GOOD	FAIR	EXCELLENT	EXCELLENT	EXCELLENT	FAIR	EXCELLENT	EXCELLENT
2 Miércoles	87	GOOD	GOOD	EXCELLENT	GOOD	FAIR	GOOD	EXCELLENT	EXCELLENT
3 Jueves	84	GOOD	FAIR	EXCELLENT	FAIR	FAIR	EXCELLENT	EXCELLENT	EXCELLENT
4 Viernes	88	GOOD	FAIR	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
5 Sábado	87	GOOD	GOOD	GOOD	EXCELLENT	GOOD	GOOD	EXCELLENT	EXCELLENT

Figura 5.22: Calificaciones de la calidad del sueño a lo largo de la semana de entrenamientos, usuario 2

Comprobamos que para el primer usuario, aunque la duración general es aceptable y el estrés muy bajo durante todos los sueños, la calidad del despertar, la inquietud y el porcentaje en las distintas zonas guarda gran peso en la calidad global del sueño.

Podemos ver cómo la mejor puntuación de sueño se alcanza la noche tras el día de recuperación entre semana, lo que posiblemente influye en el buen rendimiento del entrenamiento del jueves. La siguiente mejor puntuación también se obtiene tras un día de descanso, pero el porcentaje de sueño REM fue insuficiente.

Por otro lado, la peor calificación se obtiene la noche del viernes al sábado, que coincide con el día en que la carga de ejercicio acumulada es mayor.

El usuario 2, por su parte, presenta una gran calidad de sueño todos los días de la semana, aunque no hay evidencia de la influencia del día de descanso en este parámetro.

### 5.1.9. VO<sub>2</sub> Máximo tras la Semana de Entrenamiento

Los resultados de la Prueba de Carrera llevada a cabo por ambos usuarios tras su semana de entrenamiento reflejan que no hay evidencia de que este parámetro pueda aumentar tras una semana de actividad continua, lo que llama a un aumento de la duración del estudio a líneas futuras.

Usuario	VO <sub>2</sub> Máximo Inicial (ml/kg/min)	VO <sub>2</sub> Máximo Final (ml/kg/min)
Usuario 1	43	43
Usuario 2	46	46

Cuadro 5.15: VO<sub>2</sub> máximo inicial y obtenido tras la semana de entrenamientos

## 5.2. Análisis global. Resultados del algoritmo propuesto

En esta sección se analizan los resultados obtenidos de la ejecución del algoritmo implementado sobre los datos tras la fase de análisis. Comentaremos los **resultados de la implementación en Python** (analizando la claridad de la visualización de la tabla y los insights), el **análisis de los resultados** que podría usar un especialista para dar recomendaciones (de manera análoga al análisis seguido para cada parámetro) y, por último, la **evaluación cualitativa del algoritmo propuesto** según las sensaciones registradas de cada entrenamiento.

### 5.2.1. Resultados de la implementación del algoritmo en Python

A continuación, se muestra la salida del programa en Python para los datos de ambos usuarios.

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

En primer lugar, las tablas de calificación global de la recuperación y el rendimiento desglosadas en calificaciones individuales, resultantes son las siguientes (dividimos la tabla del usuario 2 figuras por su tamaño):

V02 Máximo actual: 43 ml/kg/min Datos de Estrés y Body Battery no disponibles													
Día	Calificación Recuperación	Calificación Calidad del Sueño	Calificación HRV	Calificación Body Battery Inicial	Calificación Recuperación Día Descanso Anterior	Calificación Rendimiento	Calificación VO2 Máximo	Calificación Frecuencia Cardíaca Media	Calificación Picos Frecuencia Cardíaca	Calificación Valles Frecuencia Cardíaca	Calificación Balance Respiración		
0 Lunes	Buena	Regular	Excelente	Excelente	-	Buena	Regular	Excelente	Regular	Excelente	Regular		
1 Martes	Buena	Regular	Buena	Bueno	-	Buena	Regular	Excelente	Buena	Excelente	Regular		
2 Miércoles	Buena	Regular	Buena	Bueno	-	-	-	-	-	-	-		
3 Jueves	Excelente	Excelente	Excelente	Excelente	Buena	Excelente	Regular	Buena	Buena	Excelente	Excelente		
4 Viernes	Buena	Regular	Buena	Bueno	-	Buena	Regular	Buena	Excelente	Excelente	Mala		
5 Sábado	Regular	Malo	Buena	Bueno	-	-	-	-	-	-	-		

Figura 5.23: Tabla de calificaciones de recuperación y rendimiento, usuario 1

V02 Máximo actual: 46 ml/kg/min													
Día	Calificación Recuperación	Calificación Calidad del Sueño	Calificación HRV	Calificación Body Battery Inicial	Calificación Body Battery Pre-Entrenamiento	Calificación Estrés Pre-Entrenamiento	Calificación Body Battery Final Día Anterior	Calificación Estrés Post Entrenamiento Día Anterior	Calificación Body Battery Final	Calificación Estrés Día Descanso	Calificación Recuperación Día Descanso Anterior	C Re	
0 Lunes	Excelente	Bueno	Excelente	Excelente	Excelente	Bueno	-	-	-	-	-	-	
1 Martes	Buena	Bueno	Excelente	Excelente	Bueno	Bueno	Bueno	Bueno	-	-	-	-	
2 Miércoles	Excelente	Bueno	Excelente	Excelente	-	-	-	-	-	Excelente	Bueno	-	
3 Jueves	Excelente	Bueno	Excelente	Excelente	Excelente	Bueno	-	-	-	-	-	Excelente	
4 Viernes	Excelente	Bueno	Excelente	Excelente	Excelente	Bueno	Excelente	Bueno	-	-	-	-	
5 Sábado	Excelente	Bueno	Excelente	Excelente	-	-	-	-	-	Excelente	Excelente	-	

Figura 5.24: Tabla de calificaciones de recuperación y rendimiento, usuario 2

Calificación Recuperación Día Descanso Anterior	Calificación Rendimiento	Calificación VO2 Máximo	Calificación Frecuencia Cardíaca Media	Calificación Picos Frecuencia Cardíaca	Calificación Valles Frecuencia Cardíaca	Calificación Balance Respiración
-	Excelente	Buena	Excelente	Buena	Buena	Buena
-	Regular	Buena	Regular	Excelente	Excelente	Excelente
-	-	-	-	-	-	-
Excelente	Regular	Buena	Mala	Excelente	Excelente	Buena
-	Regular	Buena	Excelente	Excelente	Excelente	Mala
-	-	-	-	-	-	-

Figura 5.25: Tabla de calificaciones de recuperación y rendimiento, usuario 2

Como se puede comprobar, como los datos de estrés del usuario 1 estaban corruptos, los excluimos para la calificación mediante el uso de la variable booleana (*valid\_stress = False*, por lo que tiene menos parámetros (también se muestra el mensaje)).

Por otro lado, podemos comprobar la correcta disposición en el DataFrame final de los parámetros considerados para la evaluación según el tipo de día que se evalúa: para la recuperación, se sigue perfectamente el sistema descrito en la tabla 3.6 para días de descanso (miércoles y sábado), de entrenamiento tras descanso (lunes y jueves) y de entrenamiento tras entrenamiento (martes y viernes). Para el rendimiento, podemos ver que solo se realiza la calificación los días de entrenamiento.

Las funciones de estilo se han aplicado correctamente, por lo que la visualización de la tabla es intuitiva.

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

Siguiendo con el análisis de la calidad de la interfaz de usuario construida, mostramos el conjunto de desplegables generados con los insights de información adicional de la calificación.

Aparecen plegados mostrando únicamente el título (Insights Recuperación/Rendimiento - Día) en un recuadro interactivo que, al pinchar sobre él, despliega el 'acordeón' con los comentarios:



Figura 5.26: Desplegables de insights del Lunes plegados

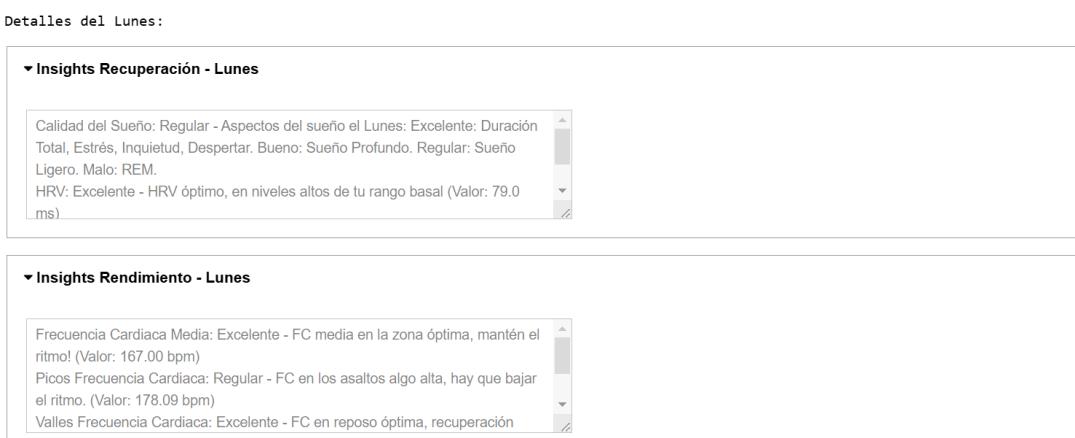


Figura 5.27: Desplegables de insights del Lunes desplegados

Los días sin entrenamiento solo imprimen los insights de la calificación de la recuperación:

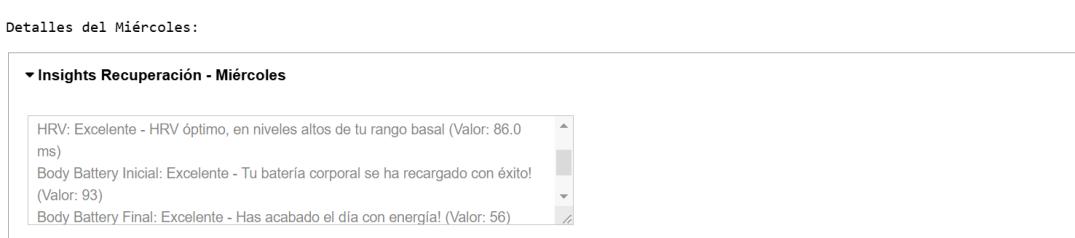


Figura 5.28: Desplegables de insights un día sin entrenamiento

### 5.2.2. Análisis de los resultados de la recuperación y el rendimiento mediante el algoritmo propuesto

En este último apartado se pretende analizar los resultados para sacar conclusiones que nos permitan relacionar el impacto de la recuperación de cara a un entrenamiento. De nuevo, este trabajo debería ser realizado por un especialista, pero se hace para describir el funcionamiento del algoritmo.

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

Analizando los resultados obtenidos para el usuario 1, podemos observar lo siguiente:

- Este registra el entrenamiento de mayor rendimiento tras la recuperación mejor valorada de la semana. Además, podemos comprobar que se produce tras un día de descanso. Los valores de energía corporal (HRV y Body Battery) también son máximos tras días de descanso.
- Por otro lado, alcanza valores negativos de algunos parámetros en el entrenamiento del viernes, y el sueño de la noche del viernes al sábado fue el peor de la semana, lo que señala los efectos negativos de la fatiga.

Por otro lado, para el usuario 2:

- El mejor resultado de rendimiento también se registra tras una día de descanso y una recuperación excelente.
- Pero por otro lado, se obtiene un mal resultado de rendimiento el otro día tras descanso de la semana, asociado a una mala calificación de la frecuencia cardíaca media, concretamente por entrenar fuera de la zona de FC óptima registrada. En este caso, el sistema de insight devuelve el siguiente comentario.

▼ Insights Rendimiento - Jueves

Frecuencia Cardiaca Media: Mala - La frecuencia media está fuera de la zona óptima, es necesario ajustar el ritmo. (Valor: 165.00 bpm)

Picos Frecuencia Cardiaca: Excelente - FC en los asaltos óptima, no hay riesgo de sobreesfuerzo! (Valor: 177.66 bpm)

Valles Frecuencia Cardiaca: Excelente - FC en reposo óptima, recuperación

Figura 5.29: Insights del entrenamiento del jueves, usuario 2

Según los resultados que devuelve el programa, el usuario de menor experiencia consigue mejores resultados que el usuario más entrenado, justificado por la mala selección de intensidad de entrenamiento del usuario 2. Además, se debe tener en cuenta que el criterio de calificación está adaptado a la capacidad cardiovascular de cada usuario y a los valores de referencia según el nivel de entrenamiento de boxeo.

De nuevo y como se comentará en el capítulo de conclusiones, estos resultados no son determinantes y no pueden tratarse con rigor, ya que el algoritmo requiere de la revisión de especialistas, de entrenamiento y de un análisis de eficiencia.

### 5.2.3. Evaluación de la eficiencia del algoritmo propuesto

Por último, se evalúa cualitativamente la eficiencia del algoritmo planteado, buscando la correlación entre los resultados obtenidos y las sensaciones y observaciones registradas de ambos usuarios de manera previa y posterior al entrenamiento (3.3.3).

Las observaciones y sensaciones registradas para cada usuario se recogen en las tablas 5.16 y 5.17, respectivamente.

Las observaciones del usuario 1 sobre su entrenamiento coinciden en gran medida con la información que nos brinda el algoritmo de calificaciones. Comentemos los resultados del jueves,

## CAPÍTULO 5. ANÁLISIS DE RESULTADOS

---

día de entrenamiento tras el día de recuperación: Al iniciar del entrenamiento del jueves, tras el día de descanso, no se sentía fatigado y se sentía con mucha energía, lo que coincide con la *excelente* calificación de la recuperación que tuvo de cara al entrenamiento de ese día. El rendimiento de dicho entrenamiento está calificado por el algoritmo como *excelente*, y el usuario comentó que aunque la intensidad fue muy alta, no se sintió tan cansado como otros días en los que entrenó a la misma intensidad, y sintió que su respiración en los asaltos de recuperación es buena (este usuario padece de asma, por lo que, como nos indicó, suele 'ahogarse' un poco tras realizar mucho esfuerzo).

	Sensaciones/Observaciones			
Aspecto	E1 (Lunes)	E2 (Martes)	E3 (Jueves)	E4 (Viernes)
Fatiga acumulada	No se sentía fatigado	Se sentía un poco fatigado	No se sentía fatigado	Se sentía un poco fatigado
Energía previa al entrenamiento	Alta	Media	Alta	Baja
Desempeño en el entrenamiento	Muy bueno	Bueno	Muy bueno	Regular, acabó muy cansado
Intensidad del entrenamiento	Alta	Adecuada	La sintió igual de alta que el primer día pero acabó menos cansado, podría haberla aumentado	Alta
Recuperación y respiración entre asaltos	Se sintió ahogado algunos asaltos	Se sintió ahogado algunos asaltos	Muy buena	Se sintió ahogado algunos asaltos

Cuadro 5.16: Sensaciones del usuario 1 durante los entrenamientos

El caso del usuario 2 es algo distinto, ya que aunque en la mayoría de los casos las calificaciones del algoritmo concuerdan con el contexto brindado por el usuario, en algún aspecto difieren.

Como podemos comprobar, el usuario no sintió fatiga acumulada a lo largo de la semana y todos los entrenamientos los empezó con energía. Esto coincide con las calificaciones de recuperación del algoritmo. Las sensaciones de respiración entre asaltos también coinciden.

Por otro lado, las observaciones de la intensidad de entrenamiento concuerdan parcialmente con los resultados del algoritmo: El entrenamiento del jueves, la calificación de FC media es mala, lo que concuerda con la descripción que da el sujeto sobre sus sensaciones (no entrenó de manera óptima), pero las obtenidas el martes y el viernes se alejan de la descripción (el martes, *regular*, sintió una buena intensidad, mientras que el viernes, *excelente*, sintió sobreesfuerzo). Esto sugiere que o el modo de calcular las zonas de frecuencia cardíaca óptima no fue correcto y se realizó sobre un valor de FC máxima en sobreesfuerzo, o que los límites de calificación/valores umbrales del algoritmo de evaluación no están elegidos con precisión. En cualquier caso, el algoritmo debería ser revisado en estudios más amplios para validar las hipótesis y los resultados encontrados.

Aspecto	Sensaciones/Observaciones			
	E1 (Lunes)	E2 (Martes)	E3 (Jueves)	E4 (Viernes)
Fatiga acumulada	No se sentía fatigado	No se sentía fatigado	No se sentía fatigado	Se sentía un poco fatigado
Energía previa al entrenamiento	Alta	Alta	Alta	Alta
Desempeño en el entrenamiento	Muy bueno	Bueno	Bueno	Regular, acabó muy cansado por el sparring, sintió mucho sobreesfuerzo
Intensidad del entrenamiento	Adecuada	Adecuada	Sintió que entrenó por debajo de sus posibilidades, no acabo cansado	Adecuada, pero muy alta durante el sparring
Recuperación y respiración entre asaltos	Buena	Buena	Buena	Acabó muy ahogado al final del entrenamiento

Cuadro 5.17: Sensaciones del usuario 2 durante los entrenamientos

Dicho esto, la alta correlación entre las calificaciones resultantes y la experiencia personal de los usuarios demuestra que el algoritmo está bien planteado y que su funcionamiento pone en práctica los fundamentos de los dispositivos wearables y de la literatura de los deportes de contacto y las ciencias de la salud revisados en el Capítulo 2, a falta de una revisión de rigor por especialistas.

### 5.3. Análisis del rendimiento de la red personal.

#### 5.3.1. Autonomía

Como se comentó en la sección 3.1.1, activar el modo de medición por segundo reduce la duración de la autonomía del smartwatch que el fabricante indica como valor por defecto.

Durante este estudio, se han llevado a cabo dos modos de carga del dispositivo, atendiendo al procedimiento que un usuario promedio podría usar:

- Una única carga completa. Referente al uso ininterrumpido del reloj, con una única carga cuando el reloj alcanza el 10 % de batería.
- Cargas diarias parciales. Se aprovechan los períodos en los que el usuario se quita el reloj (ej. duchas) para cargarlo.

Los resultados son los siguientes:

Modo de carga	Resultado de Autonomía
Carga completa	3d 17h 45m hasta el 10 %
Cargas parciales	Acabó el sexto día con 70 %

Cuadro 5.18: Autonomía del Garmin Venu 2 con el modo de medición por segundo

Como podemos ver, la medición por segundo supone una reducción a más del 50 % de la autonomía que el fabricante anunciaba para el modo de monitorización por defecto.

### 5.3.2. Valoración de la usabilidad mediante encuesta SUS

Finalmente, se muestran los resultados obtenidos de la realización de la encuesta SUS sobre el sistema de monitorización por parte de los dos sujetos del estudio, siguiendo el procedimiento descrito en la sección 3.4.

<b>Pregunta</b>	<b>Respuesta</b>	
	<b>Usuario 1</b>	<b>Usuario 2</b>
Creo que me gustaría usar este sistema con frecuencia.	4	5
Encontré el sistema innecesariamente complejo.	2	1
Pensé que el sistema era fácil de usar.	4	5
Creo que necesitaría el apoyo de un técnico para poder usar este sistema.	1	1
Encontré que las diversas funciones del sistema estaban bien integradas.	5	5
Pensé que había demasiada inconsistencia en este sistema.	4	3
Imagino que la mayoría de las personas aprenderían a usar este sistema muy rápidamente.	4	5
Encontré el sistema muy incómodo de usar.	2	1
Me sentí muy seguro(a) usando el sistema.	5	5
Necesité aprender muchas cosas antes de poder empezar a usar este sistema.	2	1
<b>Puntuación SUS</b>	72,5	92,5
<b>Calificación SUS</b>	C	A
<b>Aceptabilidad</b>	<b>Aceptable</b>	<b>Aceptable</b>

Cuadro 5.19: Encuesta SUS de la red personal de dispositivos wearables

Los resultados muestran que ambos usuarios encontraron el sistema de monitorización con dispositivos wearables poco invasivo y (portable y autónomo en cuanto a la batería) y fácil de usar (autonomía de uso). En cuanto a la fiabilidad, en términos de una buena capacidad de monitorización y conectividad, los usuarios se sintieron menos satisfechos, debido a la pérdida de datos de estrés y batería corporal diarios (un día para el usuario 2 y todos los días en el caso del usuario 1).

## 6. Conclusiones y líneas futuras

El desarrollo de este Proyecto ha permitido explorar los avances en las tecnologías de wearables en el contexto del Internet de las Cosas y cómo son capaces de optimizar aspectos de la vida de las personas, en concreto el rendimiento deportivo.

Este Trabajo ha buscado encontrar nuevas herramientas que permitan a los deportistas mejorar su práctica deportiva apoyados por la tecnología, implementando un sistema que les permita aumentar su rendimiento en el entrenamiento a partir del análisis de su recuperación, evitando posibles sobreesfuerzos que lleven a la aparición de lesiones.

Como resultado, se demuestra que es posible utilizar de forma no invasiva los dispositivos wearables para la supervisión de parámetros fisiológicos que no sólo nos permita medir y mejorar el rendimiento deportivo, si no que también conciencie a los deportistas sobre el tan importante proceso de recuperación y su impacto en los entrenamientos.

Los resultados del sistema de monitorización y el algoritmo de calificación han sido especialmente exitosos dentro de los límites alcanzables por este Proyecto, ya que demuestran que se puede mejorar la experiencia deportiva de un usuario mediante el uso de los dispositivos wearables adecuados, pero no podemos afirmar con rigor la eficacia del sistema sin el apoyo de especialistas de las ciencias de la salud y el deporte que analicen el planteamiento seguido, ni sin fases de entrenamiento y testeo del algoritmo de calificación, así como la realización estudios más profundos que cuenten con un mayor número de voluntarios.

Por otro lado, se ha explorado la rama de la captación y el análisis de datos, entendiendo la importancia de un buen procesamiento de datos en un sistema TIC y el por qué de que sean tan valiosos para las grandes compañías mundiales, y se ha implementado en el lenguaje de programación más extendido y utilizado en la actualidad, Python.

### 6.1. Análisis de los Objetivos Completados

En esta sección se analiza, a modo de autocrítica, el grado de consecución de los objetivos propuestos al inicio del Proyecto. El cumplimiento del **objetivo principal** se incluye al final de la tabla:

Objetivo	Nivel de logro	Observaciones
Explorar los distintos tipos de wearables para el despliegue de una red personal que cumpla con los requisitos de los deportes de contacto.	100 %	El profundo proceso de documentación y el estudio de mercado del sector nos permitió cumplir este objetivo.

## CAPÍTULO 6. CONCLUSIONES Y LÍNEAS FUTURAS

---

Objetivo	Nivel de logro	Observaciones
Determinar las métricas a medir por los dispositivos que brinde información valiosa sobre recuperación y rendimiento en deportes de contacto.	100 %	La exhaustiva revisión de la literatura concluyó en la caracterización del boxeo en métricas medibles.
Evaluar que los wearables utilizados sean poco invasivos y tengan la capacidad de monitorización adecuada al caso de estudio.	100 %	El procedimiento experimental, complementado por la opinión de los usuarios mediante la encuesta de usabilidad SUS nos permitió evaluar estas características fundamentales de la tecnología wearable.
Analizar y establecer estrategias de análisis de los datos para brindar información valiosa que pudiera ser utilizada por especialistas.	100 %	En este Proyecto se planteó un sistema de monitorización, se desarrolló un sistema de depurado y análisis de los datos captados, así como una propuesta de algoritmo de calificación que permitiría a especialistas realizar recomendaciones a partir del análisis.
Aplicar el sistema desarrollado y el procedimiento de analítica de datos.	100 %	El sistema experimental de monitorización se implementó con éxito, así como, en Python, el bloque de depurado y análisis de los datos y el algoritmo de calificación propuesto.
Realizar un piloto con un conjunto amplio de sujetos para validar el sistema y algoritmos propuestos.	10 %	La limitación de tiempo y recursos ha impedido el cumplimiento total de este objetivo, por lo que queda como pendiente para una investigación futura.
Utilizar buenas prácticas de desarrollo de software para el proceso de desarrollo y puesta a punto de la infraestructura empleada para el sistema de supervisión.	100 %	Todo el proceso de desarrollo técnico realizado durante el Proyecto queda bien documentado y ha seguido buenas prácticas.
<b>Objetivo principal del Proyecto. Monitorización del rendimiento deportivo y la recuperación de un practicante de deportes de contacto mediante el despliegue de una red personal de dispositivos wearables y la explotación de los datos capturados mediante su análisis.</b>	80 %	Mediante el cumplimiento de los objetivos específicos listados en esta tabla, se ha logrado con éxito el planteamiento y el despliegue de un sistema de monitorización de la actividad en deportes de contacto, demostrando cómo los dispositivos wearables son capaces de mejorar la experiencia de usuario en este ámbito. Sin embargo, no se ha podido evaluar el algoritmo propuesto con un conjunto de usuarios lo suficientemente amplio y, además, el sistema de monitorización propuesto requeriría una revisión de rigor por especialistas de las ciencias del deporte y la salud. Por tanto, la consecución total del objetivo principal de este Proyecto queda pendiente de investigaciones futuras.

Cuadro 6.1: Tabla de nivel de consecución de los objetivos propuestos

## 6.2. Líneas Futuras

- **Mejora de la precisión del análisis.** El trabajo conjunto con un equipo de especialistas de las ciencias de la salud y el deporte, permitiría mejorar el sistema de monitorización planteado, ajustando los valores de referencia a cada usuario con un respaldo de rigor.
- **Mejora del algoritmo de calificación.** El uso de técnicas de entrenamiento basadas aprendizaje automático y machine learning, permitiría mejorar la precisión del algoritmo de calificación planteado, también de la mano de especialistas de la salud.
- **Comparación con datos clínicos.** Para evaluar la eficacia del sistema, los puntos anteriores podrían complementarse con la comparación de las métricas medidas por wearables y calculadas con nuestro sistema con las medidas con equipo clínico certificado.
- **Ampliación del alcance del estudio y el tiempo de monitorización.** Aumentar el número de participantes en el estudio permitiría tener un mayor número de perfiles, lo que permitiría obtener resultados más concluyentes.  
Además, si se ampliara el span de tiempo de monitorización de cada usuario, tendríamos las muestras necesarias para un análisis de mayor precisión.
- **Integración en una aplicación móvil.** A nivel de visualización, sería muy atractivo desarrollar una aplicación móvil que permita al usuario acceder a sus datos de una manera simple, a través de una interfaz de usuario intuitiva.
- **Desarrollo de una aplicación Garmin Connect IQ.** Por otro lado, se propone continuar con el desarrollo de la aplicación en el smartwatch, en el caso de que Garmin amplíe el conjunto de variables a las que pueden acceder desarrolladores terceros.

## 6.3. Conclusiones personales

A nivel personal, el desarrollo de este Proyecto me ha podido llevar a cabo una aplicación del Internet de las Cosas a un caso de estudio que me ha acompañado este último año y que, aunque está poco investigado, ha demostrado tener efectos muy positivos en la salud, no solo física, sino también mental en el caso de niños y adolescentes víctimas de bullying y acoso escolar. Además, me ha permitido familiarizarme con el análisis de datos en Python, sector en el que he decidido enfocar mi carrera profesional.



# Bibliografía

- [1] M. Slimani, H. Znazen, M. Sellami, and P. Davis, “Heart rate monitoring during combat sports matches: a brief review,” vol. 18, no. 2, pp. 273–292, publisher: Routledge eprint: <https://doi.org/10.1080/24748668.2018.1469080>. [Online]. Available: <https://doi.org/10.1080/24748668.2018.1469080>
- [2] Smartwatch: previsión de ventas mundiales 2016-2025. [Online]. Available: <https://es.statista.com/estadisticas/664393/prevision-de-las-ventas-mundiales-de-smartwatches/>
- [3] Worldwide wearables market is forecast to rebound in 2023 with continued growth thereafter, according to IDC. [Online]. Available: <https://www.idc.com/getdoc.jsp?containerId=prUS50984723>
- [4] H. Simões, P. M. Santos, B. Pereira, and A. Figueiredo, “As artes marciais e os desportos de combate e o bullying: uma revisão sistemática (martial arts and combat sports and the bullying: a systematic review) (las artes marciales y deportes de combate y el acoso escolar: una revisión sistemática),” *Retos*, vol. 39, pp. 834–843, ene 2021.
- [5] E. Sazonov, *Wearable Sensors: Fundamentals, Implementation and Applications*. Academic Press, google-Books-ID: F9PVDwAAQBAJ.
- [6] C. G. C. Golondrino, M. A. O. Alarcón, and M. E. M. Ríos, “Arquitectura IoT para el desarrollo de sistemas de monitorización y análisis de variables fisiológicas en el área de asistencia médica,” vol. 8, no. 3, pp. 1–13, number: 3 Publisher: Universidad Simón Bolívar. [Online]. Available: <https://revistas.unisimon.edu.co/index.php/innovacioning/article/view/4699>
- [7] C. M. de Boxeo, *Reglamento WBC Amateur*. WBC, 2016.
- [8] F. E. de Kickboxing, *Reglamento De Kickboxing Amateur*. FEK, 2017.
- [9] El boxeo: sistema de competición, características fisiológicas, antropométricas y hábitos dietético-nutricionales. [Online]. Available: <https://www.efdeportes.com/efd167/el-boxeo-habitos-dietetico-nutricionales.htm>
- [10] Valoración del rendimiento en boxeo, [1 propuesta]. Section: Deportes de combate. [Online]. Available: <https://mundoentrenamiento.com/valoracion-del-rendimiento-en-boxeo/>
- [11] EstiloMMA - blog de boxeo, muay thai, MMA y BBJ. [Online]. Available: <https://www.estilomma.com/blog/beneficios-de-saltar-a-la-comba-en-boxeo-n-107.html>
- [12] Boxeo, las lesiones más frecuentes - fisiolution. [Online]. Available: <https://fisiolution.com/boxeo-las-lesiones-mas-frecuentes/>
- [13] Rodrigo merlo biomecánica. [Online]. Available: <https://www.cie-dc.com/rodrigo-merlo-biomecaacutenica.html>

## BIBLIOGRAFÍA

---

- [14] EstiloMMA - blog de boxeo, muay thai, MMA y BBJ. [Online]. Available: <https://www.estilomma.com/blog/por-que-los-boxeadores-entrenan-el-cuello-n-163.html?srsltid=AfmBOop-IWC67VEADZsu7p-2KAsywiu-m5aRv2pg2jG1lBk0Q7xwRj-s>
- [15] L. Guidetti, A. Musulin, and C. Baldari, "Physiological factors in middleweight boxing performance," *Journal of Sports Medicine and Physical Fitness*, vol. 42, no. 3, pp. 309–314, 2002.
- [16] F. Silvi, M. Andriana, I. Wardhani, Y. dian p, I. Narasinta, and S. Melaniani, "The effect of boxing exergame on blood lactate and VO<sub>2max</sub> in non-athlete healthy young men," vol. 11, pp. 827–831.
- [17] E. Arseneau, S. Mekary, and L. A. Léger, "Vo<sub>2</sub> requirements of boxing exercises," vol. 25, no. 2, p. 348. [Online]. Available: [https://journals.lww.com/nsca-jscr/FullText/2011/02000/V\\_o2\\_Requirements\\_of\\_Boxing\\_Exercises.9.aspx?casa\\_token=IjuPHDzV8IkAAAAAA:5557q3ou4oFNDxpBpQAhItSONxAD0pk0qEKyARKF9u3qXCOJlsjdyGFXIIQbvOE7-TxQs3KDOUhY6E1Ir1HPxzUQ0vi-AxdSA](https://journals.lww.com/nsca-jscr/FullText/2011/02000/V_o2_Requirements_of_Boxing_Exercises.9.aspx?casa_token=IjuPHDzV8IkAAAAAA:5557q3ou4oFNDxpBpQAhItSONxAD0pk0qEKyARKF9u3qXCOJlsjdyGFXIIQbvOE7-TxQs3KDOUhY6E1Ir1HPxzUQ0vi-AxdSA)
- [18] V. Bruzas, A. Stasiulis, A. Cepulenas, P. Mockus, B. Statkeviciene, and V. Subacius, "Aerobic capacity is correlated with the ranking of boxers," *Perceptual and Motor Skills*, vol. 119, no. 1, pp. 50–58, aug 2014.
- [19] P. Joko, "Physical and physiological characteristics of hungarian boxers," *Europe Boxing Magazine*, vol. 8, pp. 28–29, 1983.
- [20] A. K. Ghosh, A. Goswami, and A. Ahuja, "Heart rate and blood lactate response in amateur competitive boxing," *Indian Journal of Medical Research*, vol. 102, pp. 179–183, 1995.
- [21] G. L. Khanna and I. Manna, "Study of physiological profile of indian boxers," *Journal of Sports Science and Medicine*, vol. 5, no. CSSI, pp. 90–98, 2006.
- [22] M. Smith, "Physiological profile of senior and junior england international amateur boxer," *Journal of Sports Science and Medicine*, vol. 5, no. CSSI, pp. 74–89, 2006.
- [23] A. K. Ghosh, "Heart rate, oxygen consumption and blood lactate responses during specific training in amateur boxing," *International Journal Applications Sports Sciences*, vol. 22, pp. 1–12, 2010.
- [24] A.-C. Macquet and V. Skalej, "Time management in elite sports: How do elite athletes manage time under fatigue and stress conditions?" vol. 88, no. 2, pp. 341–363, eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/joop.12105>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/joop.12105>
- [25] J. Taelman, S. Vandeput, A. Spaepen, and S. Van Huffel, "Influence of mental stress on heart rate and heart rate variability," in *4th European Conference of the International Federation for Medical and Biological Engineering*, J. Vander Sloten, P. Verdonck, M. Nyssen, and J. Haueisen, Eds. Springer, pp. 1366–1369.
- [26] Frecuencia cardiaca de reserva e intensidades en el entrenamiento de resistencia. Grupo Sobre Entrenamiento (G-SE). Accedido: 18 de octubre de 2024. [Online]. Available: <https://g-se.com/frecuencia-cardiaca-de-reserva-e-intensidades-en-el-entrenamiento-de-resistencia-bp-v57cfb26da71be>
- [27] P. Davis, R. M. Leithauser, and R. Beneke, "The energetics of semicontact 3 × 2-min amateur boxing," *International Journal of Sports Physiology and Performance*, vol. 9, pp. 233–239, 2014.

## BIBLIOGRAFÍA

---

- [28] L. F. De Lira Peixinho-Pena, R. L. Vancini, R. J. De Freitas Guina Fachina, A. A. De Almeida, M. d. S. Andrade, and A. C. Da Silva, “Heart rate response during a simulated olympic boxing match is predominantly above ventilatory threshold 2: A cross-sectional study,” *Open Access Journal Sports Medicine*, vol. 4, pp. 175–182, 2013.
- [29] P. T. Nikolaidis, F. M. Clemente, K. Busko, and B. Knechtle, “Physiological responses to simulated boxing: The effect of sitting versus standing body position during breaks: A pilot study,” *Asian Journal of Sports Medicine*, vol. 8, no. 3, p. e55434, 2017.
- [30] C. López. Frecuencia cardiaca y entrenamiento. [Online]. Available: <https://fundaciondelcorazon.com/ejercicio/calculo-y-monitorizacion/3161-frecuencia-cardiaca-y-entrenamiento.html>
- [31] Y. Tran, N. Wijesuriya, M. Tarvainen, P. Karjalainen, and A. Craig, “The relationship between spectral changes in heart rate variability and fatigue,” vol. 23, no. 3, pp. 143–151, publisher: Hogrefe Publishing. [Online]. Available: <https://econtent.hogrefe.com/doi/10.1027/0269-8803.23.3.143>
- [32] P. L. Valenzuela, G. Sánchez-Martínez, E. Torrontegi, J. Vázquez-Carrión, Z. Montalvo, and O. Kara, “Validity, reliability, and sensitivity to exercise-induced fatigue of a customer-friendly device for the measurement of the brain’s direct current potential,” *Journal of Strength and Conditioning Research*, 2020.
- [33] C. Schneider, F. Hanakam, T. Wiewelhove, A. Döweling, M. Kellmann, T. Meyer, M. Pfeiffer, and A. Ferrauti, “Heart rate monitoring in team sports—a conceptual framework for contextualizing heart rate measures for training and recovery prescription,” *Frontiers in Physiology*, vol. 9, 2018.
- [34] M. Buchheit, “Monitoring training status with hr measures: Do all roads lead to rome?” *Frontiers in Physiology*, vol. 5, 2014.
- [35] K. Peterson, “Resting heart rate variability can predict track and field sprint performance,” vol. 1, pp. 1–9.
- [36] J. Morales, J. M. Álamo, X. García-Massó, B. Buscà, J. L. López, P. Serra-Añó, and L.-M. González, “Use of heart rate variability in monitoring stress and recovery in judo athletes,” vol. 28, no. 7, p. 1896. [Online]. Available: [https://journals.lww.com/nsca-jscr/fulltext/2014/07000/use\\_of\\_heart\\_rate\\_variability\\_in\\_monitoring\\_stress.14.aspx](https://journals.lww.com/nsca-jscr/fulltext/2014/07000/use_of_heart_rate_variability_in_monitoring_stress.14.aspx)
- [37] C. Urzeala, A. Bota, S. Serbanouiu, M. Mezei, F. Dutheil, and D. Courteix, “Heart rate variability as a possible predictor of sport performance in junior rhythmic gymnastics,” vol. 28, no. 2, pp. 171–179, publisher: IOS Press. [Online]. Available: <https://content-iospress-com/articles/isokinetics-and-exercise-science/ies192222>
- [38] Monitor de energía body battery™ — tecnología garmin. Accedido: 26 de septiembre de 2024. [Online]. Available: <https://www.garmin.com/es-ES/garmin-technology/health-science/body-battery/>
- [39] S. Pérez, “INFLUENCIA DE LA RESPIRACIÓN EDUCADA MEDIANTE SUGESTIÓN, SOBRE LA RECUPERACIÓN DEL LUCHADOR, CATEGORÍA ESCOLAR.”
- [40] M. Slimani, A. H. Paravlic, H. Chaabene, P. Davis, K. Chamari, and F. Cheour, “Hormonal responses to striking combat sports competition: a systematic review and meta-analysis,” vol. 35, no. 2, pp. 121–136, publisher: Termedia. [Online]. Available: <https://www.termedia.pl/Hormonal-responses-to-striking-combat-sports-competition-r-na-systematic-review-and-meta-analysis,78,31035,0,1.html>

## BIBLIOGRAFÍA

---

- [41] El estrés y la salud cardíaca. [Online]. Available: <https://www.goredforwomen.org/es/healthy-living/healthy-lifestyle/stress-management/stress-and-heart-health>
- [42] N. S. Simpson, E. L. Gibbs, and G. O. Matheson, “Optimizing sleep to maximize performance: Implications and recommendations for elite athletes,” *Scandinavian Journal of Medicine & Science in Sports*, vol. 27, no. 3, pp. 266–274, 2017.
- [43] G. W. Kirschen, J. J. Jones, and L. Hale, “The impact of sleep duration on performance among competitive athletes,” *Clinical Journal of Sport Medicine*, vol. 1, 2018.
- [44] P. J. Arnal, T. Lapole, M. Erblang, M. Guillard, C. Bourrilhon, D. Léger, and G. Y. Millet, “Sleep extension before sleep loss: Effects on performance and neuromuscular function,” *Medicine and Science in Sports and Exercise*, vol. 48, no. 8, pp. 1595–1603, 2016.
- [45] HRV status | garmin technology. [Online]. Available: <https://www.garmin.com/en-US/garmin-technology/health-science/hrv-status/>
- [46] C. Cardio. ¿qué es la arritmia sinusal? CampusCardio. Accedido: 2 de octubre de 2024. [Online]. Available: <https://campuscardio.com/arritmia-sinusal/>
- [47] P. Astrand and K. Rodhal, *Text book of work physiology. Physiological bases of exercise*, 3rd ed. New York: McGraw-Hill Book Co., 1977.
- [48] J. O. C. Coyne, A. J. Coutts, R. Fomin, D. N. French, R. U. Newton, and G. G. Haff, “Heart rate variability and direct current measurement characteristics in professional mixed martial arts athletes,” vol. 8, no. 8, p. 109, number: 8 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: <https://www.mdpi.com/2075-4663/8/8/109>
- [49] Monitorización de la frecuencia cardiaca | ciencias de la salud | tecnología garmin | garmin. [Online]. Available: <https://www.garmin.com/es-ES/garmin-technology/health-science/heart-rate-monitoring/>
- [50] Pulse ox | garmin technology. [Online]. Available: <https://www.garmin.com/es-ES/garmin-technology/health-science/pulse-ox/>
- [51] Connect iq sdk — garmin developers. Accedido: 26 de septiembre de 2024. [Online]. Available: <https://developer.garmin.com/connect-iq/overview/>
- [52] Polar h10 | polar españa. [Online]. Available: <https://www.polar.com/es/sensors/h10-heart-rate-sensor>
- [53] Garmin and G. L. o. i. subsidiaries. Garmin HRM-pro™ plus | monitor de frecuencia cardiaca. [Online]. Available: <https://www.garmin.com/es-ES/p/770963>
- [54] Anaconda — the operating system for ai. Anaconda. Accedido: 26 de abril de 2024. [Online]. Available: <https://www.anaconda.com/>
- [55] Project jupyter. Accedido: 26 de abril de 2024. [Online]. Available: <https://jupyter.org>
- [56] os — Miscellaneous operating system interfaces. Python documentation. Accedido: 17 de mayo de 2024. [Online]. Available: <https://docs.python.org/3/library/os.html>
- [57] json — JSON encoder and decoder. Python documentation. Accedido: 6 de mayo de 2024. [Online]. Available: <https://docs.python.org/3/library/json.html>
- [58] re — Regular expression operations. Python documentation. Accedido: 17 de mayo de 2024. [Online]. Available: <https://docs.python.org/3/library/re.html>

## BIBLIOGRAFÍA

---

- [59] Pyplot tutorial — Matplotlib 3.9.2 documentation. Matplotlib documentation. Accedido: 7 de agosto de 2024. [Online]. Available: <https://matplotlib.org/stable/tutorials/pyplot.html>
- [60] find\_peaks — SciPy v1.14.1 Manual. SciPy documentation. Accedido: 15 de septiembre de 2024. [Online]. Available: [https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find\\_peaks.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html)
- [61] DataFrame — pandas 2.2.3 documentation. Accedido: 6 de mayo de 2024. [Online]. Available: <https://pandas.pydata.org/docs/reference/frame.html>
- [62] A. Casero. Cómo eliminar duplicados con sets en python. Accedido: 17 de mayo de 2024. [Online]. Available: <https://keepcoding.io/blog/eliminar-duplicados-con-sets-en-python/>
- [63] Widget List — Jupyter Widgets 8.1.5 documentation. Accedido: 2 de octubre de 2024. [Online]. Available: <https://ipywidgets.readthedocs.io/en/8.1.5/examples/Widget%20List.html>
- [64] Símbolos, tutoriales y recursos para escribir en latex. Accedido: 9 de octubre de 2024. [Online]. Available: <https://manualdelatex.com/>
- [65] Sorting Techniques. Python documentation. Accedido: 17 de mayo de 2024. [Online]. Available: <https://docs.python.org/3/howto/sorting.html>



# A. Implementación del depurado de datos en Python

## A.0.1. Módulos y librerías empleados

Los módulos y librerías empleados para la limpieza de los datos, explicados en la sección 3.2.1, se listan a continuación:

Listing A.1: Módulos y librerías importados para el depurado de datos

```
import os
import json
import pandas as pd
```

Listing A.1: Módulos y librerías importados para el depurado de datos

## Disposición de los datos

Se crearon dos carpetas para el almacenado de datos con objetivo de su limpieza. Una en la que se encuentran los datos brutos a leer y otra en la que se almacenarán los datos una vez procesados.

Listing A.2: Directorios para el depurado de datos

```
#Carpetas de entrada y salida
carpeta_brutos = "C:/Users/marco/Downloads/
                    datosBrutosTFG/usuario1"
carpeta_procesados = "C:/Users/marco/Downloads/datosProcesadosTFG/usuario1"
```

Listing A.2: Directorios para el depurado de datos

## Funciones empleadas

El funcionamiento de los métodos definidos para el depurado de los datos se explica a continuación:

- **convertir\_a\_segundos(hora\_str):** Función a la que se le introduce como parámetro de entrada un string con el formato horario de 24 h, "HH:MM", devolviendo la cantidad de segundos desde las 00:00 h que le corresponde.

Se empleará para la eliminación de datos de estrés durante el entrenamiento.

- **eliminar\_duplicados(data):** Esta función hace una distinción dependiendo de si el parámetro a extraer tiene estructura de lista (frecuencia cardíaca durante el entrenamiento) o de diccionario (valores continuos en archivos de Resumen de Salud).

Se ha implementado la técnica de eliminado de duplicados con *sets* de Python [62]: Se crea la lista *resultado* en la que se almacenará la salida ya procesada, además de un *set*, una estructura de Python que no permite dos muestras iguales, que almacena la marca temporal (o el índice, si es diccionario) de la muestra evaluada si aún no la contiene. En ese caso, el valor se almacena en la lista *resultado*.

- **rellenar\_faltantes(data, step=1):** Función que recoge como parámetro de entrada un conjunto de datos (en forma de diccionario, ya que es la estructura de los archivos que la usan) y una variable *step* que establece cada cuantos segundos (o pasos) se recogen datos.

En primer lugar se pasan las 'claves' del índice del diccionario a enteros y se reordenan, permitiéndonos iterar secuencialmente y detectar así cualquier hueco.

Se crea un diccionario con los resultados y se abre un bucle en el que, para el rango de índices, avanzando según en pasos de valor *step*, se comprueba si para esa muestra hay valor y si no se rellena con la actual. Fijémonos en como los índices se operan como strings (*str[i]*), ya que corresponden a las claves de los diccionarios.

- **procesar\_entrenamiento(data):** Esta función recoge datos de los archivos de entrenamiento y llama a la función *eliminar\_duplicados* para la lista *samples*, que almacena la FC durante el entrenamiento.

Por otro lado, solo se añaden los valores de FC a la lista *resultado* si están dentro del rango establecido, evitando valores anómalos.

No llama a la función relleno de datos faltantes ya que la captura de datos, al ser mediante una aplicación (Barra) se produce de manera irregular en el tiempo.

- **procesar\_resumen\_salud(data):** Esta función recoge datos de los archivos de resumen de salud y llama a la función *eliminar\_duplicados* y *rellenar\_faltantes* para el diccionario *epochSummaries* en *summaries*, que almacena los parámetros capturados por segundo.

- **procesar\_estres(data):** Esta función recoge datos de los archivos de estrés y llama a la función *eliminar\_duplicados* y *rellenar\_faltantes* para los diccionarios *timeOffsetStressLevelValues* y *timeOffsetBodyBatteryValues* en *timeOffsetValues*, que almacena los parámetros capturados por segundo.

Se llama cuando el archivo a evaluar no corresponde con un día de entrenamiento.

- **procesar\_estres\_entrenamiento(data, start\_time, duration):**

Esta versión de la función se llama cuando el archivo evaluado corresponde a un día de entrenamiento.

A parte del procedimiento explicado de su versión anterior, en esta función se realiza la eliminación de los datos de estrés durante el entrenamiento, para lo que recoge como entrada el tiempo (en segundos) en el que empieza y su duración. Estos tiempos se recogen (en horas, por lo que se implementó la función *convertir\_a\_segundos*) en el diccionario *entrenamientos*, definido manualmente a partir de los datos de Garmin Connect.

Se hace un bucle en el que la marca temporal y su valor iteran desde que comienza el entrenamiento hasta que acaba, comprobando si hay valores no nulos y, en ese caso, anulándolos.

Listing A.3: Funciones para el depurado de datos

```
#Diccionario donde se especifican manualmente los entrenamientos para los
#archivo de estres
entrenamientos = {
```

```
"reloj_36_health_stress_20241007.json": ("19:45", "01:36"),
"reloj_36_health_stress_20241008.json": ("19:03", "01:36"),
"reloj_36_health_stress_20241010.json": ("19:52", "01:28"),
"reloj_36_health_stress_20241011.json": ("18:07", "01:31"),
"reloj_36_health_stress_20241014.json": ("17:37", "01:26"),
"reloj_36_health_stress_20241015.json": ("17:33", "01:15"),
"reloj_36_health_stress_20241017.json": ("17:57", "01:15"),
"reloj_36_health_stress_20241018.json": ("17:35", "01:21"),
}

def convertir_a_segundos(hora_str):
    #Pasamos el formato de hora:minuto a segundos
    horas, minutos = map(int, hora_str.split(":"))
    return horas * 3600 + minutos * 60

def eliminar_duplicados(data):
    #Funcion para eliminar duplicados, dependiendo de si la estructura del JSON
    #es lista o diccionario.
    if isinstance(data, list):
        resultado = []
        tiempos_previos = set() #Usamos la estructura set para evitar
        #duplicados
        for muestra in data:
            if muestra["startTimeInSeconds"] not in tiempos_previos:
                resultado.append(muestra)
                tiempos_previos.add(muestra["startTimeInSeconds"])
        return resultado
    elif isinstance(data, dict):
        resultado = {}
        indices_previos = set()
        for indice, valor in data.items():
            if indice not in indices_previos:
                resultado[indice] = valor
                indices_previos.add(indice)
        return resultado

def llenar_faltantes(data, step=1):
    #Funcion que rellena los valores faltantes con el valor anterior conocido
    indices = sorted(map(int, data.keys()))
    resultado = {}

    for i in range(indices[0], indices[-1] + step, step):
        if str(i) in data:
            resultado[str(i)] = data[str(i)]
        else:
            resultado[str(i)] = resultado[str(i - step)]

    return resultado

def procesar_entrenamiento(data):
    #Archivos de entrenamiento. Elimina duplicados y descarta valores FC fuera
```

```
    del rango (40, 220).
for item in data:
    if "samples" in item:
        item["samples"] = eliminar_duplicados(item["samples"])

        item["samples"] = [
            muestra for muestra in item["samples"]
            if 40 <= muestra.get("heartRate", 0) <= 220
        ]
return data

def procesar_resumen_salud(data):
    #Archivos de Resumen de Salud. Elimina duplicados y rellena faltantes.
    for item in data:
        for summary in item.get("summaries", []):
            if "epochSummaries" in summary:
                summary["epochSummaries"] = eliminar_duplicados(summary["epochSummaries"])
                summary["epochSummaries"] = rellenar_faltantes(summary["epochSummaries"], step=1)
    return data

def procesar_estres(data):
    #Archivos de Estres y Body Battery. Elimina duplicados y rellena faltantes.
    end_time = start_time + duration

    for item in data:
        item["timeOffsetStressLevelValues"] = eliminar_duplicados(item["timeOffsetStressLevelValues"])
        item["timeOffsetStressLevelValues"] = rellenar_faltantes(item["timeOffsetStressLevelValues"], step=180)

        item["timeOffsetBodyBatteryValues"] = eliminar_duplicados(item["timeOffsetBodyBatteryValues"])
        item["timeOffsetBodyBatteryValues"] = rellenar_faltantes(item["timeOffsetBodyBatteryValues"], step=180)

    return data

def procesar_estres_entrenamiento(data, start_time, duration):
    #Archivos de Estres y Body Battery. Elimina duplicados y rellena faltantes.
    end_time = start_time + duration

    for item in data:
        item["timeOffsetStressLevelValues"] = eliminar_duplicados(item["timeOffsetStressLevelValues"])
        item["timeOffsetStressLevelValues"] = rellenar_faltantes(item["timeOffsetStressLevelValues"], step=180)

        item["timeOffsetBodyBatteryValues"] = eliminar_duplicados(item["timeOffsetBodyBatteryValues"])
```

```
item["timeOffsetBodyBatteryValues"] = rellenar_faltantes(item["timeOffsetBodyBatteryValues"], step=180)

#Si hay valores de estres durante el entrenamiento, los cambiamos por -1
for timestamp, valor in item["timeOffsetStressLevelValues"].items():
    timestamp = int(timestamp)
    if start_time <= timestamp <= end_time and valor > 0:
        item["timeOffsetStressLevelValues"][str(timestamp)] = -1

return data
```

Listing A.3: Funciones para el depurado de datos

Finalmente, en el bucle principal se recorre la carpeta de datos brutos y se van cargando, ejecutándose la función de procesado indicada según el nombre del archivo que se está evaluando y si es día de entrenamiento (para el caso de los archivos de estrés):

Listing A.4: Bucle principal para el depurado de datos

```
#Bucle principal. Llamada a funciones y gestion de directorios (lectura y guardado)
for filename in os.listdir(carpeta_brutos):
    #Leemos los datos brutos
    if filename.endswith(".json"):
        with open(os.path.join(carpeta_brutos, filename), "r", encoding="utf-8") as file:
            data = json.load(file)

        if "health_stress" in filename and filename in entrenamientos:
            #Convertimos el tiempo de inicio y duracion de formato "HH:MM" a segundos
            start_time_str, duration_str = entrenamientos[filename]
            start_time = convertir_a_segundos(start_time_str)
            duration = convertir_a_segundos(duration_str)

            data = procesar_estres_entrenamiento(data, start_time, duration)
        elif "health_stress" in filename and filename not in entrenamientos:
            data = procesar_estres(data)
        elif "activity_detail" in filename:
            data = procesar_entrenamiento(data)
        elif "health_snapshot" in filename:
            data = procesar_resumen_salud(data)

        #Guardamos los datos ya procesados
        procesados = os.path.join(carpeta_procesados, filename)
        with open(procesados, "w", encoding="utf-8") as outfile:
            json.dump(data, outfile, ensure_ascii=False, indent=4)
```

Listing A.4: Bucle principal para el depurado de datos



## B. Código complementario para la implementación del análisis de datos en Python

### B.0.1. Gestión de directorios

Las funciones empleadas para identificar estos índices y ordenar las carpetas con dicho criterio son las siguientes, concretamente para el caso de los archivos de entrenamiento [65]:

Listing B.1: Funciones de gestión de directorios

```
##Gestion de Archivos##
#Ruta a la carpeta en la que se encuentran los archivos de entrenamiento
carpeta_json = 'C:/Users/marco/Downloads/datosTFG/usuario1/Entrenamiento'

#Funcion para extraer el numero del nombre del archivo
def obtener_numero(archivo):
    return int(re.search(r'(\d+)\.json$', archivo).group(1))

#Ordena los archivos segun el numero obtenido con la funcion anterior
archivos_json = sorted([archivo for archivo in os.listdir(carpeta_json) if
                        archivo.endswith('.json')], key=obtener_numero)
```

Listing B.1: Funciones de gestión de directorios

- En primer lugar se especifica la ruta en la que se encuentran los archivos y se almacena en la variable '`carpeta_json`'.
- La función '`obtener_numero`' usa un patrón de búsqueda con la función `search` del módulo `re`. El patrón es `r'(\d+)\.json$'`, donde `\d+` busca uno o más dígitos consecutivos y `\.json$` especifica que esos dígitos están antes de la extensión '.json'. Por último, con `group(1)` se especifica que se extrae lo que está dentro de los paréntesis, en este caso los dígitos.
- La función `sorted` ordena esta lista usando `listdir` del módulo `os`, con el criterio impuesto por `obtener_numero`, especificado en el parámetro `key`.

### B.0.2. Funciones de análisis de parámetros

#### Frecuencia cardíaca

Listing B.2: Variables iniciales para frecuencia cardiaca

```
#Listas y variables iniciales
nombres_dias = ['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sabado']
    #Para la columna del DF que asocia los valores a un dia de la semana
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```
dias_entrenamiento = ['Lunes', 'Martes', 'Jueves', 'Viernes'] #Para los labels de las graficas (solo dias de entrenamiento)
hr_values = [] #Lista en la que almacenaremos los resultados medios, minimos y maximos de frecuencia cardiaca por dia
contador_entrenamiento = 0 #Contador que usamos para el numero de entrenamiento para el titulo de las graficas
```

Listing B.2: Variables iniciales para frecuencia cardiaca

Listing B.3: Representación de la frecuencia cardíaca durante el entrenamiento

```
##Grafica de FC a lo largo del entrenamiento##
plt.figure(figsize=(10, 5))
plt.plot(heart_rates_flat, marker='.', markersize=1, color='r')

#FC Media
plt.plot(mean_hr, marker='.', markersize=1, color='b')
plt.annotate(f'Med:', xy=(len(heart_rates_flat) - 1, hr_medio),
            xytext=(len(heart_rates_flat) - 400, hr_medio - 5), color='b')
plt.annotate(f'{hr_medio}', xy=(len(heart_rates_flat) - 1, hr_medio),
            xytext=(len(heart_rates_flat) - 400, hr_medio - 9), color='b')

#FC max y min
plt.scatter(max_index, max_hr, color='g', s=10, zorder=3, label='Max_Value',
            ) # Destacar el punto maximo
plt.annotate(f'Max:{max_hr}', color='g', xy=(max_index, max_hr), xytext=(max_index + 10, max_hr + 1))

plt.scatter(min_index, min_hr, color='y', s=10, zorder=3, label='Min_Value',
            ) # Destacar el punto minimo
plt.annotate(f'Min:{min_hr}', color='y', xy=(min_index, min_hr), xytext=(min_index + 10, min_hr - 3))

plt.title(f'Frecuencia_cardiaca_durante_el_entrenamiento_(Entrenamiento_{contador_entrenamiento+1})')
plt.xlabel('Tiempo(s)')
plt.ylabel('Frecuencia_Cardiaca(bpm)')
plt.grid(True)
plt.tight_layout()
plt.show()

contador_entrenamiento += 1
```

Listing B.3: Representación de la frecuencia cardíaca durante el entrenamiento

Listing B.4: DF y graficas de evolucion frecuencia cardiaca

```
#Creamos DF con valores de FC
hr_df_temp = pd.DataFrame(hr_values)

#Anadimos filas con los dias nulos
dias_nulos = {'Dia': ['Miercoles', 'Sabado'], 'hr_medio': [None, None], 'max_hr': [None, None], 'min_hr': [None, None]}
dias_nulos_df = pd.DataFrame(dias_nulos)
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```
#Concatenamos
hr_df = pd.concat([hr_df_temp, dias_nulos_df], ignore_index=True)
hr_df['Dia'] = pd.Categorical(hr_df['Dia'], categories=nombres_dias, ordered=True)
hr_df = hr_df.sort_values('Dia').reset_index(drop=True)

#Filtramos para quedarnos con dias de entrenamiento para las graficas
df_solo_entrenamiento = hr_df_completo.dropna(subset=['hr_medio', 'max_hr', 'min_hr'])

#Grafica de evolucion de FC media, max y min
plt.figure(figsize=(10, 6))
plt.plot(df_solo_entrenamiento['Dia'], df_solo_entrenamiento['hr_medio'], marker='o', color='b', label='Frecuencia_Cardiaca_Media')
plt.plot(df_solo_entrenamiento['Dia'], df_solo_entrenamiento['max_hr'], marker='o', color='r', label='Frecuencia_Cardiaca_Maxima')
plt.plot(df_solo_entrenamiento['Dia'], df_solo_entrenamiento['min_hr'], marker='o', color='y', label='Frecuencia_Cardiaca_Minima')
plt.xticks(rotation=45, ha='right')
plt.title('Evolucion_de_la_FC_Media,_Maxima_y_Minima_por_Dia')
plt.xlabel('Dia')
plt.ylabel('Frecuencia_Cardiaca_(bpm)')
plt.tight_layout()
plt.legend()
plt.show()
```

Listing B.4: DF y graficas de evolucion frecuencia cardiaca

### Picos y valles de frecuencia cardíaca

Listing B.5: Variables iniciales para picos de frecuencia cardiaca

```
#Diccionarios, listas y variables iniciales.
#Especificamos los picos y valles a filtrar manualmente para cada archivo
#segun su indice (impreso mas adelante)
picos_a_eliminar_manual = {
    'EntrenamientoBoxeo_1.json': [3222, 5714, 2702],
    'EntrenamientoBoxeo_2.json': [2871, 3578, 5740],
    'EntrenamientoBoxeo_3.json': [],
    'EntrenamientoBoxeo_4.json': [4319]
}

valles_a_eliminar_manual = {
    'EntrenamientoBoxeo_1.json': [1, 353, 609, 844, 1909, 2984, 4215],
    'EntrenamientoBoxeo_2.json': [7, 441, 607, 2534, 3717],
    'EntrenamientoBoxeo_3.json': [3, 274, 552, 793, 1840, 2623, 3584, 3810,
        4122, 4324],
    'EntrenamientoBoxeo_4.json': [29, 399, 757, 1046, 1479, 1843, 3439]
}
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```
nombres_dias = ['Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sábado']
días_entrenamiento = ['Lunes', 'Martes', 'Jueves', 'Viernes']
picos_medios = []
valles_medios = []
contador_entrenamiento = 0
```

Listing B.5: Variables iniciales para picos de frecuencia cardíaca

Listing B.6: Representación de picos y valles en la gráfica de frecuencia cardíaca durante el entrenamiento

```
#Grafica de picos y valles para cada entrenamiento
plt.figure(figsize=(10, 5))
plt.plot(heart_rates_flat, color='lightgray', label='Frecuencia_Cardiaca')

plt.scatter(picots, [heart_rates_flat[p] for p in picots], color='r', label='Picos', zorder=3)
plt.scatter(valles, [heart_rates_flat[v] for v in valles], color='b', label='Valles', zorder=3)

plt.axhline(y=pd.Series([heart_rates_flat[p] for p in picots]).mean(), color='r', linestyle='--', label='Media_Picos')
plt.axhline(y=pd.Series([heart_rates_flat[v] for v in valles]).mean(), color='b', linestyle='--', label='Media_Valles')

plt.annotate(f'Med_Picos:{picots_media:.2f}', xy=(len(heart_rates_flat) - 1, picots_media),
            xytext=(-150, picots_media + 5), color='r')
plt.annotate(f'Med_Valles:', xy=(len(heart_rates_flat) - 1, valles_media),
            xytext=(len(heart_rates_flat) - 400, valles_media - 5), color='b')
plt.annotate(f'{valles_media:.2f}', xy=(len(heart_rates_flat) - 1,
            valles_media),
            xytext=(len(heart_rates_flat) - 400, valles_media - 9), color='b')

plt.title(f'Picos_y_valles_de_FC_del_entrenamiento_(Entrenamiento_{contador_entrenamiento+1})')
plt.xlabel('Tiempo_(s)')
plt.ylabel('Frecuencia_Cardiaca_(bpm)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Listing B.6: Representación de picos y valles en la gráfica de frecuencia cardíaca durante el entrenamiento

Listing B.7: DF de picos de frecuencia cardíaca

```
#DF picos y valles
df_picots_valles_temp = pd.DataFrame({
    'Dia': días_entrenamiento,
    'Picos_Medios': picos_medios,
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```
'Valles_Medios': valles_medios
})
#Anadimos dias sin entrenamiento
dias_nulos = {'Dia': ['Miercoles', 'Sabado'], 'Picos_Medios': [None, None], 'Valles_Medios': [None, None]}
dias_nulos_df = pd.DataFrame(dias_nulos)

#Concatenamos
df_picos_valles = pd.concat([df_picos_valles_temp, dias_nulos_df], ignore_index=True)
df_picos_valles['Dia'] = pd.Categorical(df_picos_valles['Dia'], categories=nombres_dias, ordered=True)
df_picos_valles = df_picos_valles.sort_values('Dia').reset_index(drop=True)
```

Listing B.7: DF de picos de frecuencia cardiaca

Listing B.8: Grafica de evolucion de picos de frecuencia cardíaca

```
#Nos quedamos con los dias con entrenamiento
df_picos_valles_filtrado = df_picos_valles_completo.dropna(subset=['Picos_Medios', 'Valles_Medios'])
etiquetas_entrenamientos = df_picos_valles_filtrado['Dia']

#Grafica de evolucion de media de picos y valles
plt.figure(figsize=(10, 6))

plt.plot(etiquetas_entrenamientos, df_picos_valles_filtrado['Picos_Medios'], marker='o', color='r', label='Picos_Medios')
plt.plot(etiquetas_entrenamientos, df_picos_valles_filtrado['Valles_Medios'], marker='o', color='b', label='Valles_Medios')

plt.title('Evolucion del valor medio de picos y valles de FC de los entrenamientos')
plt.xlabel('Entrenamiento')
plt.ylabel('Frecuencia Cardiaca Media (bpm)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Listing B.8: Grafica de evolucion de picos de frecuencia cardíaca

## HRV

Listing B.9: Variables iniciales para HRV

```
#Listas y variables iniciales
dias = ['dia1', 'dia2', 'dia3', 'dia4', 'dia5', 'dia6'] #nombre de las carpetas
nombres_dias = ['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sabado'] #para el campo comun del DF
colores = ['r', 'g', 'b', 'c', 'm', 'y']
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```
horas_x = [8, 14, 19, 21, 24] #Para la escala del eje X de la grafica de evolucion diaria
todas_horas_x = []
etiquetas_x = ['Por_la_manana', 'Medio_dia', 'Pre-Entreno', 'Post-Entreno', 'Por_la_noche'] #Etiquetas para las graficas de evolucion diaria
todos_rmssd_values = []
```

Listing B.9: Variables iniciales para HRV

Listing B.10: DF de HRV

```
#DF con HRV RMSSD
rmssd_df = pd.DataFrame(todos_rmssd_values, columns=etiquetas_x[:len(todos_rmssd_values[0])])
rmssd_df['Dia'] = nombres_dias[:len(rmssd_df)]
rmssd_df = rmssd_df[['Dia']] + etiquetas_x[:len(todos_rmssd_values[0])]
```

Listing B.10: DF de HRV

Listing B.11: Representación de la evolución diaria del HRV

```
## Dentro del Bucle Principal de HRV ##

#Grafica de evolucion diaria para cada dia
plt.figure(figsize=(10, 6))
plt.plot(horas_actuales, rmssd_values, marker='o', linestyle='-', color=colores[idx], label=f'Evolucion_{nombres_dias[idx]}')
plt.xticks(horas_x, etiquetas_x, rotation=45)
plt.title(f'Evolucion del HRV_RMSSD_{nombres_dias[idx]}')
plt.xlabel('Tiempo_(Momento_del_dia)')
plt.ylabel('HRV_RMSSD_(ms)')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

Listing B.11: Representación de la evolución diaria del HRV

Listing B.12: Representación de la comparación del HRV diario a lo largo de la semana

```
#Grafica comparativa del HRV de todos los dias.
plt.figure(figsize=(10, 6))

for idx, (rmssd_values, horas_actuales) in enumerate(zip(todos_rmssd_values, todas_horas_x)):
    plt.plot(horas_actuales, rmssd_values, marker='o', linestyle='-', color=colores[idx], label=nombres_dias[idx])

plt.xticks(horas_x, etiquetas_x, rotation=45)
plt.title('Comparacion del HRV_RMSSD_entre_dias')
plt.xlabel('Tiempo_(Momento_del_dia)')
plt.ylabel('HRV_RMSSD_(ms)')
plt.grid(True)
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```
plt.legend()  
plt.tight_layout()  
plt.show()
```

Listing B.12: Representación de la comparación del HRV diario a lo largo de la semana

### Estrés y balance de respiración pre/post entrenamiento

Listing B.13: Variables iniciales para FR y estres pre/post entrenamiento

```
#Diccionarios, listas y variables iniciales  
#Archivos a analizar. Se indican los días de entrenamiento y los archivos  
correspondientes a los RdS pre y post entrenamiento.  
entrenamientos = {  
    'dia1': (3, 4),  
    'dia2': (8, 9),  
    'dia4': (16, 17),  
    'dia5': (21, 22)  
}  
  
#Volcado manual de la respiración media durante el entrenamiento  
respiracion_durante_entrenamiento = {  
    'dia1': 19,  
    'dia2': 19,  
    'dia4': 18,  
    'dia5': 19  
}  
  
dias = ['dia1', 'dia2', 'dia3', 'dia4', 'dia5', 'dia6']  
nombres_dias = ['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sabado']  
datos_respiracion = []  
datos_estres = []
```

Listing B.13: Variables iniciales para FR y estres pre/post entrenamiento

Listing B.14: Bucle de extracción de datos de estrés y respiración pre/post entrenamiento

```
#Bucle principal  
for idx, dia in enumerate(dias):  
    nombre_dia = nombres_dias[idx]  
  
    #Gestion de directorios  
    carpeta_dia = os.path.join(carpeta_principal, dia)  
  
    ##Funcion de ordenacion de archivos para cada dia##  
  
    if dia in entrenamientos: #Los días de entrenamiento dividimos entre pre y  
        post_entrenamiento  
        pre_entrenamiento_idx, post_entrenamiento_idx = entrenamientos[dia] #Se  
            declara para dicho dia los archivos pre y post segun el diccionario  
  
        ## Carga de datos ## Se indica que archivos de la lista buscamos, para  
            cargar los datos diferenciando entre pre y post
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```
archivos_necesarios = [f'{pre_entrenamiento_idx}:02d}', f'{post_entrenamiento_idx}:02d}']
archivos_presentes = [archivo for archivo in archivos_json if any(f'_{num}.json' in archivo for num in archivos_necesarios)]
```

#Funcion que carga dos datos de estres y respiracion segun nombre del archivo, nos permite cargar diferenciando entre pre y post

```
def cargar_datos_pre_post(archivo):
    with open(os.path.join(carpeta_dia, archivo), 'r') as f:
        data = json.load(f)
    if isinstance(data, list) and 'summaries' in data[0]:
        summaries_df = pd.DataFrame(data[0]['summaries'])
        respiracion = summaries_df.loc[summaries_df['summaryType'] == 'respiration', 'avgValue'].values[0]
        estres = summaries_df.loc[summaries_df['summaryType'] == 'stress', 'avgValue'].values[0]
        return respiracion, estres
    else:
        print(f"Advertencia: No se encontraron datos de 'summaries' en {archivo}")
        return None, None
```

respiracion\_pre\_value, estres\_pre\_value = cargar\_datos\_pre\_post(archivos\_presentes[0])
respiracion\_post\_value, estres\_post\_value = cargar\_datos\_pre\_post(archivos\_presentes[1])

#Dias de entrenamiento

```
if respiracion_pre_value is not None and respiracion_post_value is not None:
    datos_respiracion.append({
        'Dia': nombre_dia,
        'Respiracion_Pre': respiracion_pre_value,
        'Respiracion_Durante': respiracion_durante_entrenamiento[dia],
        'Respiracion_Post': respiracion_post_value,
        'Balance_Respiracion': respiracion_post_value -
            respiracion_pre_value
    })
    datos_estres.append({
        'Dia': nombre_dia,
        'Estres_Pre': estres_pre_value,
        'Estres_Post': estres_post_value
    })
else:
    #Dias sin entrenamiento
    datos_respiracion.append({
        'Dia': nombre_dia,
        'Respiracion_Pre': None,
        'Respiracion_Durante': None,
        'Respiracion_Post': None,
        'Balance_Respiracion': None
    })
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```
    })
    datos_estres.append({
        'Dia': nombre_dia,
        'Estres_Pre': None,
        'Estres_Post': None
    })

#DFs con datos de respiracion y estres
df_respiracion = pd.DataFrame(datos_respiracion)
df_estres = pd.DataFrame(datos_estres)
```

Listing B.14: Bucle de extracción de datos de estrés y respiración pre/post entrenamiento

Listing B.15: Representación de la FR y el estrés pre y post entrenamiento

```
#Nos quedamos con dias de entrenamiento
df_solo_entrenamiento = df_respiracion.dropna(subset=['Respiracion_Pre', 'Respiracion_Durante', 'Respiracion_Post'])
df_estres_solo_entrenamiento = df_estres[df_estres['Dia'].isin(df_solo_entrenamiento['Dia'])]

# Grafica de Respiracion: Pre-entrenamiento, Entrenamiento y Post-entrenamiento
etiquetas_x = [nombre_dia for nombre_dia in df_solo_entrenamiento['Dia']]
plt.figure(figsize=(10, 6))
plt.bar(df_solo_entrenamiento.index - 0.2, df_solo_entrenamiento['Respiracion_Pre'], width=0.2, label='Pre-entrenamiento', color='b')
plt.bar(df_solo_entrenamiento.index, df_solo_entrenamiento['Respiracion_Durante'], width=0.2, label='Entrenamiento', color='g')
plt.bar(df_solo_entrenamiento.index + 0.2, df_solo_entrenamiento['Respiracion_Post'], width=0.2, label='Post-entrenamiento', color='r')
plt.xticks(df_solo_entrenamiento.index, etiquetas_x, rotation=45)
plt.title('Frecuencia Respiratoria Media Pre, Durante y Post Entrenamiento')
plt.ylabel('Frecuencia Respiratoria (rpm)')
plt.xlabel('Entrenamiento')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# Grafica de Estres: Pre-entrenamiento y Post-entrenamiento
plt.figure(figsize=(10, 6))
plt.bar(df_estres_solo_entrenamiento.index - 0.1, df_estres_solo_entrenamiento['Estres_Pre'], width=0.2, label='Pre-entrenamiento', color='b')
plt.bar(df_estres_solo_entrenamiento.index + 0.1, df_estres_solo_entrenamiento['Estres Post'], width=0.2, label='Post-entrenamiento', color='r')
plt.xticks(df_estres_solo_entrenamiento.index, etiquetas_x, rotation=45)
plt.title('Nivel de Estres Medio Pre y Post Entrenamiento')
plt.ylabel('Nivel de Estres')
plt.xlabel('Entrenamiento')
plt.legend()
plt.grid(True)
plt.tight_layout()
```

```
plt.show()
```

Listing B.15: Representación de la FR y el estrés pre y post entrenamiento

### Estrés y batería corporal

Listing B.16: Variables iniciales para estrés y Body Battery

```
#Diccionarios y listas para almacenar variables
entrenamientos = { #Hora inicio y duracion de los entrenamientos
    "Lunes": ("17:37", "01:26"),
    "Martes": ("17:33", "01:15"),
    "Jueves": ("17:57", "01:15"),
    "Viernes": ("17:35", "01:21"),
}
dias_semana = ['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sabado']
estres_dias = []
body_battery_dias = []
```

Listing B.16: Variables iniciales para estrés y Body Battery

Listing B.17: Primera parte del bucle de extracción y representacion de datos de estres y Body Battery.

```
#Bucle principal
for idx, archivo in enumerate(archivos_json):

    dia_semana = dias_semana[idx % len(dias_semana)] #Recorremos dias de la
    semana con idx

    ##Carga de Archivos##

    #Inicializamos valores como None, ya que los datos estan corruptos, para
    #anadirlos al DF
    estres_pre = None
    estres_post = None
    estres_medio_descanso = None
    max_battery_value = None
    max_battery_time = None
    last_battery_value = None
    last_battery_time = None
    battery_pre_entrenamiento = None

    if isinstance(data, list) and len(data) > 0:
        estres_data = data[0]
        time_offset_body_battery = estres_data.get('timeOffsetBodyBatteryValues',
            {})
        time_offset_stress = estres_data.get('timeOffsetStressLevelValues', {})

        #DF con mapeos estres/BB-timeOffset
        df_body_battery = pd.DataFrame(list(time_offset_body_battery.items()),
            columns=['TimeOffset', 'BodyBattery'])
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```

df_body_battery['TimeOffset'] = df_body_battery['TimeOffset'].astype(
    int) / 3600 #Pasamos a horas

df_stress = pd.DataFrame(list(time_offset_stress.items()), columns=[,
    'TimeOffset', 'StressLevel'])
df_stress['TimeOffset'] = df_stress['TimeOffset'].astype(int) / 3600
df_stress = df_stress[df_stress['StressLevel'] >= 0] #Filtramos valores nulos

#Valores maximos y finales de BB
max_battery_time = df_body_battery.loc[df_body_battery['BodyBattery'].idxmax()['TimeOffset']]
max_battery_value = df_body_battery['BodyBattery'].max()
last_battery_value = df_body_battery['BodyBattery'].iloc[-1]
last_battery_time = df_body_battery['TimeOffset'].iloc[-1]

#Almacenamos la BB en listas
body_battery_dias.append({
    'Dia': dia_semana,
    'Body_Battery_Max': max_battery_value,
    'Hora_Max': max_battery_time,
    'Body_Battery_Final': last_battery_value,
    'Hora_Final': last_battery_time,
    'Body_Battery_Pre-Entrenamiento': battery_pre_entrenamiento
})

```

Listing B.17: Primera parte del bucle de extracción y representación de datos de estrés y Body Battery.

La batería corporal se representa como una línea continua a lo largo del tiempo (plot), para el estrés, por otro lado, usamos la función *iterrows()* de *Pandas*. Esta función itera por las filas (de valores) del DF de manera sencilla, imprimiéndose una linea vertical en cada valor.

Listing B.18: Representación de la batería corporal junto al estrés diario

```

#Grafica de evolucion diaria de estres y BB
plt.figure(figsize=(10, 6))

#BB continuo max y ultimo
plt.plot(df_body_battery['TimeOffset'], df_body_battery['BodyBattery'],
          color='gray', linewidth=1, label='Body_Battery')
plt.scatter(max_battery_time, max_battery_value, color='red', zorder=5,
            label=f'Max:{max_battery_value}')
plt.scatter(last_battery_time, last_battery_value, color='red', zorder=5,
            label=f'Ultimo:{last_battery_value}')
if battery_pre_entrenamiento is not None:
    plt.scatter(inicio_entrenamiento, battery_pre_entrenamiento, color=
        'purple', zorder=5, label=f'BB_Pre:{battery_pre_entrenamiento}')
    plt.text(inicio_entrenamiento, battery_pre_entrenamiento - 4, f'Pre:{battery_pre_entrenamiento}', color='purple', fontsize=10, ha='center')
    plt.text(max_battery_time, max_battery_value + 2, f'Max:{
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```
    max_battery_value}', color='red', fontsize=10, ha='center')
plt.text(last_battery_time, last_battery_value - 4, f'Ultimo:{last_battery_value}', color='red', fontsize=10, ha='center')

#Estres con barras verticales
for _, row in df_stress.iterrows():
    color = 'orange' if row['StressLevel'] > 25 else 'blue'
    plt.axvline(x=row['TimeOffset'], ymin=0, ymax=row['StressLevel'] / 100, color=color, alpha=0.5)

plt.title(f'BodyBattery y Estres-{dia_semana}')
plt.xlabel('Tiempo(Horas)')
plt.ylabel('BodyBattery / Nivel de Estres')
plt.grid(True)
plt.tight_layout()
plt.legend()
plt.show()
```

Listing B.18: Representación de la batería corporal junto al estrés diario

Listing B.19: DF de estrés y Body Battery

```
#DFs de estres y BB
df_estres = pd.DataFrame(estres_dias)
df_body_battery = pd.DataFrame(body_battery_dias)
```

Listing B.19: DF de estrés y Body Battery

### Balance calórico

Listing B.20: Variables iniciales para uso calorico

```
#Diccionario y listas para almacenar variables
dias_semana = ['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sabado']
calorias_totales = []
calorias_reposo = []
calorias_activas = []
calorias_entrenos = []

#Volcado manual de las calorías de los entrenamientos
calorias_entrenamiento = { # 1:Lunes, 2:Martes, 4:Jueves, 5:Viernes
    1: 1266,
    2: 1196,
    4: 1025,
    5: 1043
}
```

Listing B.20: Variables iniciales para uso calorico

Listing B.21: Bucle de extraccion de datos de gasto calorico

```
#Bucle que itera sobre los archivos
for archivo in archivos_json:
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```
##Carga de Archivos##  
  
# Verificar que el archivo contiene las calorías  
if isinstance(data, list) and len(data) > 0:  
    calorias_data = data[0]  
  
    #Extraccion de datos  
    active_calories = calorias_data.get('activeKilocalories', 0)  
    bmr_calories = calorias_data.get('bmrKilocalories', 0)  
    calorias_totales_dia = active_calories + bmr_calories  
  
    #Almacenado en las listas  
    calorias_totales.append(calorias_totales_dia)  
    calorias_activas.append(active_calories)  
    calorias_reposo.append(bmr_calories)  
  
    #Anadimos las calorías de entrenamiento si es un dia con entrenamiento  
    dia_indice = obtener_numero(archivo)  
  
    if dia_indice in calorias_entrenamiento:  
        calorias_entrenos.append(calorias_entrenamiento[dia_indice])  
    else:  
        calorias_entrenos.append(0) # si no hay entrenamiento, calorías de  
        #entrenamiento = 0
```

Listing B.21: Bucle de extraccion de datos de gasto calorico

Listing B.22: DF de gasto calorico

```
#DF de gasto calorico  
calorias_total = {  
    'Dia': dias_semana,  
    'Calorias_Totales': calorias_totales,  
    'Calorias_Reposo' : calorias_reposo,  
    'Calorias_Activas': calorias_activas,  
    'Calorias_Entrenamiento': calorias_entrenos  
}  
df_calorias = pd.DataFrame(calorias_total)
```

Listing B.22: DF de gasto calorico

La representación del gasto calórico se hace con un diagrama de barras superpuestas apilando por día el consumo de calorías diferenciado por tipo de actividad (BMR, actividad y entrenamiento).

Listing B.23: Representación del gasto calórico diario

```
#Grafica de gasto calorico  
plt.figure(figsize=(10, 6))  
  
bar_width = 0.4  
indice_barras = range(len(df_calorias))  
  
plt.bar(indice_barras, df_calorias['Calorias_Totales'], bar_width, label='  
    Calorias_Totales', color='lightblue')
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```
plt.bar(indice_barras, df_calorías['Calorías_Activas'], bar_width, label='Calorías_Activas', color='blue', alpha=0.7)
plt.bar(indice_barras, df_calorías['Calorías_Etrenamiento'], bar_width, label='Calorías_Etrenamiento', color='red', alpha=0.7)

plt.xticks(indice_barras, df_calorías['Dia'])
plt.title('Gasto_Calórico_Diario')
plt.xlabel('Dia_de_la_Semana')
plt.ylabel('Calorías_(kcal)')
plt.legend()
plt.tight_layout()
plt.show()
```

Listing B.23: Representación del gasto calórico diario

### Calidad del sueño

Listing B.24: Variables iniciales para el análisis de la calidad del sueño

```
#Listas para almacenar variables
días_semana = ['Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sábado']
calificaciones_días = []
```

Listing B.24: Variables iniciales para el análisis de la calidad del sueño

Listing B.25: Bucle de extracción y representación de datos de sueño

```
#Bucle principal
for idx, archivo in enumerate(archivos_json):

    ##Carga de Archivos##

    #Obtenemos (si hay) los datos de sueño almacenados en un mapa de niveles
    if isinstance(data, list) and len(data) > 0:
        sleep_data = data[0]
        sleep_levels_map = sleep_data.get('sleepLevelsMap', {})

        #Mapeamos los niveles al tiempo de sueño con la función previa
        tiempos, niveles = procesar_fases_sueño(sleep_levels_map)

    plt.figure(figsize=(10, 6))

    for (start, end), nivel in zip(tiempos, niveles):
        #Rellenamos los segmentos correspondientes a cada zona a modo de barras
        plt.fill_between([start, end], 0, nivel, color='lightblue' if nivel == 1 else ('b' if nivel == 2 else 'purple'))

    plt.yticks([1, 2, 3], ['Ligero', 'Profundo', 'REM'])
    dia_semana = días_semana[idx % len(días_semana)]
```

## APÉNDICE B. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ANÁLISIS DE DATOS EN PYTHON

```
plt.title(f'Fases del Sueño - {dia_semana}')
plt.xlabel('Tiempo (segundos desde inicio del sueño)')
plt.ylabel('Fases del sueño')
plt.grid(True)
plt.tight_layout()
plt.show()

#Extraccion de calificaciones
overall_score = sleep_data.get('overallSleepScore', {}).get('value',
    None)
overall_qualifier = sleep_data.get('overallSleepScore', {}).get(
    'qualifierKey', None)
sleep_scores = sleep_data.get('sleepScores', {})

#Almacenamiento en la lista
calificaciones_dias.append({
    'Dia': dias_semana[idx % len(dias_semana)],
    'Puntuacion Global': overall_score,
    'Calificacion Global': overall_qualifier,
    'Duracion Total': sleep_scores.get('totalDuration', {}).get(
        'qualifierKey', None),
    'Estres': sleep_scores.get('stress', {}).get('qualifierKey', None),
    'Inquietud': sleep_scores.get('restlessness', {}).get('qualifierKey',
        None),
    'Despertar': sleep_scores.get('awakeCount', {}).get('qualifierKey',
        None),
    'REM': sleep_scores.get('remPercentage', {}).get('qualifierKey',
        None),
    'Sueno Ligero': sleep_scores.get('lightPercentage', {}).get(
        'qualifierKey', None),
    'Sueno Profundo': sleep_scores.get('deepPercentage', {}).get(
        'qualifierKey', None)
})
}
```

Listing B.25: Bucle de extraccion y representacion de datos de sueño

Listing B.26: Construccion de tabla de calificaciones de sueño

```
#DF con las calificaciones de sueño
df_sueno = pd.DataFrame(calificaciones_dias)

#Visualizacion con mapeo de colores
styled_df = df_sueno.style.applymap(color_cell, subset=['Calificacion Global',
    'Duracion Total', 'Estres', 'Despertar', 'REM', 'Inquietud', 'Sueno Ligero',
    'Sueno Profundo'])
styled_df
```

Listing B.26: Construccion de tabla de calificaciones de sueño



## C. Código complementario para la implementación del algoritmo de calificación en Python

### C.0.1. Funciones de evaluación de los parámetros de evaluación

Listing C.1: 'Diccionarios de Insights de recomendaciones para los distintos parámetros

```
#Diccionarios de insights por calificacion para cada parametro
insights_dict_fc = {
    'Excelente': {
        'hr_medio': "FC\u201cmedia\u201c\u201en\u201ela\u201czona\u201coptima,\u201cmanten\u201cel\u201c\u201crito\u201d!",
        'Picos\u201cMedios': "FC\u201cen\u201clos\u201casaltos\u201coptima,\u201chno\u201chay\u201c\u201criesgo\u201d\u201de\u201csobreesfuerzo!\u201d",
        'Valles\u201cMedios': "FC\u201cen\u201creposo\u201coptima,\u201c\u201crecuperacion\u201coptima!\u201d"
    },
    'Buena': {
        'hr_medio': "FC\u201cmedia\u201ccerca\u201cdel\u201ela\u201czona\u201coptima,\u201cgenial\u201cpero\u201caun\u201cpuedes\u201cmejorar!\u201d",
        'Picos\u201cMedios': "FC\u201cen\u201clos\u201casaltos\u201ccontrolada,\u201crendimiento\u201caceptable.\u201d",
        'Valles\u201cMedios': "FC\u201cen\u201creposo\u201ccontrolada,\u201c\u201crecuperacion\u201cfavorable.\u201d"
    },
    'Regular': {
        'hr_medio': "FC\u201cmedia\u201cuera\u201d\u201del\u201ela\u201czona\u201coptima,\u201cintenta\u201cbajar\u201cel\u201c\u201crito\u201d."
        ,
        'Picos\u201cMedios': "FC\u201cen\u201clos\u201casaltos\u201calgo\u201calta,\u201chay\u201c\u201chay\u201c\u201cbajar\u201cel\u201c\u201crito\u201d.",
        'Valles\u201cMedios': "FC\u201cen\u201creposo\u201calgo\u201calta,\u201chay\u201c\u201chay\u201c\u201cbajar\u201cel\u201c\u201crito\u201d."
    },
    'Mala': {
        'hr_medio': "La\u201cfrecuencia\u201cmedia\u201cesta\u201fuera\u201d\u201del\u201ela\u201czona\u201coptima,\u201ches\u201cnecesario\u201cajustar\u201cel\u201c\u201crito\u201d.",
        'Picos\u201cMedios': "FC\u201cen\u201clos\u201casaltos\u201cmuy\u201calta,\u201chay\u201c\u201chay\u201c\u201criesgo\u201d\u201de\u201csobreesfuerzo.\u201d",
        'Valles\u201cMedios': "FC\u201cen\u201creposo\u201cmuy\u201calta,\u201c\u201crecuperacion\u201centre\u201casaltos\u201chno\u201cfavorable.\u201d"
    }
}

insights_dict_hrv = {
    'Excelente': {'HRV': "HRV\u201coptimo,\u201chen\u201nniveles\u201caltos\u201d\u201detu\u201crango\u201c\u201cbasal\u201d"}, 
    'Buena': {'HRV': "HRV\u201cbueno,\u201cdentro\u201detu\u201crango\u201c\u201cbasal\u201d"}, 
    'Regular': {'HRV': "HRV\u201cdesbalanceado,\u201choy\u201centrena\u201con\u201ccalma.\u201d"}, 
    'Mala': {'HRV': "HRV\u201cbajo,\u201chay\u201c\u201crecomienda\u201descansar\u201choy.\u201d"}
}
```

## APÉNDICE C. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ALGORITMO DE CALIFICACIÓN EN PYTHON

```
insights_dict_resp = {
    'Excelente': {'Balance_Respiracion': "El balance respiratorio es excelente, indica buena regulación."},
    'Buena': {'Balance_Respiracion': "El balance respiratorio es bueno, cerca del nivel óptimo."},
    'Regular': {'Balance_Respiracion': "El balance respiratorio está algo elevado, puede ser necesario relajarse más."},
    'Mala': {'Balance_Respiracion': "El balance respiratorio está muy alto, sugiere tensión o fatiga."}
}

insights_dict_estres = {
    'Excelente': {
        'Estres_Pre-Entrenamiento': "Estres pre-entrenamiento bajo, preparación óptima.",
        'Estres_Post-Entrenamiento': "Estres post-entrenamiento bajo, recuperación óptima.",
        'Estres_Medio_Dia_Descanso': "El nivel de estres tu día de descanso ha sido óptimo.",
        'Estres_Post_Entrenamiento_Dia_Anterior': "El estres tras el entrenamiento del día anterior fue bajo, recuperación óptima."
    },
    'Bueno': {
        'Estres_Pre-Entrenamiento': "Estres pre-entrenamiento moderado, preparación adecuada.",
        'Estres_Post-Entrenamiento': "Estres post-entrenamiento moderado, recuperación aceptable.",
        'Estres_Medio_Dia_Descanso': "El nivel de estres tu día de descanso ha sido aceptable.",
        'Estres_Post_Entrenamiento_Dia_Anterior': "El estres tras el entrenamiento del día anterior fue moderado, recuperación adecuada."
    },
    'Regular': {
        'Estres_Pre-Entrenamiento': "Estres pre-entrenamiento elevado, preparación deficiente.",
        'Estres_Post-Entrenamiento': "Estres post-entrenamiento elevado, recuperación insuficiente.",
        'Estres_Medio_Dia_Descanso': "El nivel de estres en dia de descanso es algo alto, intenta relajarte.",
        'Estres_Post_Entrenamiento_Dia_Anterior': "El estres tras el entrenamiento del día anterior fue alto, recuperación deficiente."
    },
    'Malo': {
        'Estres_Pre-Entrenamiento': "Estres pre-entrenamiento muy alto, mala preparación.",
        'Estres_Post-Entrenamiento': "Estres post-entrenamiento muy elevado, recuperación deficiente.",
        'Estres_Medio_Dia_Descanso': "El nivel de estres en dia de descanso fue muy alto, se recomienda relajación."
    }
}
```

## APÉNDICE C. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ALGORITMO DE CALIFICACIÓN EN PYTHON

```

        'Estres_Post_Entrenamiento_Dia_Anterior': "El estres tras el
        entrenamiento del dia anterior fue muy alto, recuperacion pobre."
    }
}

insights_dict_body_battery = {
    'Excelente': {
        'Body_Battery_Max': "Tu bateria corporal se ha recargado con exito!",
        'Body_Battery_Final': "Has acabado el dia con energia!",
        'Body_Battery_Pre-Entrenamiento': "Tienes energia para afrontar el
        entrenamiento!"
    },
    'Bueno': {
        'Body_Battery_Max': "Nivel maximo de body battery bueno, recarga
        adecuada.",
        'Body_Battery_Final': "Body battery final en buen estado.",
        'Body_Battery_Pre-Entrenamiento': "Nivel de body battery al inicio del
        entrenamiento es adecuado."
    },
    'Regular': {
        'Body_Battery_Max': "Nivel maximo de body battery regular, podria
        mejorar la recarga.",
        'Body_Battery_Final': "Body battery final regular, posible fatiga.",
        'Body_Battery_Pre-Entrenamiento': "Nivel de body battery al inicio del
        entrenamiento es regular, podria afectar el rendimiento."
    },
    'Malo': {
        'Body_Battery_Max': "Nivel maximo de body battery bajo, indica falta de
        recarga.",
        'Body_Battery_Final': "Body battery final muy bajo, alto riesgo de
        fatiga.",
        'Body_Battery_Pre-Entrenamiento': "Nivel de body battery bajo al inicio
        del entrenamiento, rendimiento comprometido."
    }
}

```

Listing C.1: 'Diccionarios de Insights de recomendaciones para los distintos parámetros'

Listing C.2: 'Función de evaluación de VO2 Máximo'

```

#VO2 Maximo
def evaluar_vo2(vo2):

    if vo2 > 54.5:
        calificacion = 'Excelente'
    elif vo2 > 45.4:
        calificacion = 'Buena'
    elif vo2 > 41.7:
        calificacion = 'Regular'
    else:
        calificacion = 'Mala'

    resultado = pd.DataFrame([

```

## APÉNDICE C. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ALGORITMO DE CALIFICACIÓN EN PYTHON

```
'VO2_Maximo': vo2,
'Calificacion_VO2': calificacion
}])
return resultado

resultados_vo2 = evaluar_vo2(VO2_Max)
```

Listing C.2: 'Función de evaluación de VO2 Máximo'

Listing C.3: 'Función de evaluación del HRV'

```
#Evaluacion del HRV
df_hrv_reducido = rmssd_df[['Dia', 'Por_la_manana']]

def evaluar_hrv(df, insights_dict, HRV_Basal):
    resultados = []

    hrv_medio = sum(HRV_Basal) / 2 #Necesitamos el valor medio del rango basal
                                    para la calificacion

    for index, row in df.iterrows():
        dia = row['Dia']
        hrv_valor = row['Por_la_manana']

        #Evaluacion de HRV
        if pd.notna(hrv_valor):
            if HRV_Basal[0] <= hrv_valor <= HRV_Basal[1]:
                if hrv_valor >= hrv_medio:
                    calificacion_hrv = 'Excelente'
                else:
                    calificacion_hrv = 'Buena'
            else:
                diff_hrv = min(abs(hrv_valor - HRV_Basal[0]), abs(hrv_valor -
                    HRV_Basal[1]))
                if diff_hrv <= 10:
                    calificacion_hrv = 'Regular'
                else:
                    calificacion_hrv = 'Mala'
            else:
                calificacion_hrv = 'Mala'

            comentario_hrv = f"{insights_dict[calificacion_hrv]['HRV']} (Valor:{hrv_valor} ms)"

            #Creamos diccionario con calificaciones e insights
            resultado_dia = {
                'Dia': dia,
                'Calificacion_HRV': calificacion_hrv,
                'Insight_HRV': comentario_hrv
            }
            resultados.append(resultado_dia)
        #Devolvemos DF
    return pd.DataFrame(resultados)
```

## APÉNDICE C. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ALGORITMO DE CALIFICACIÓN EN PYTHON

```
#Llamamos a la funcion
resultados_hrv = evaluar_hrv(df_hrv_reducido, insights_dict_hrv, HRV_Basal)
```

Listing C.3: 'Función de evaluación del HRV'

La función de evaluación del sueño tiene una estructura distinta, ya que el DF resultante del análisis ya cuenta con las calificaciones. Esto es porque Garmin incluye las calificaciones en sus archivos JSON de salida.

Esta función en primer lugar traduce las calificaciones al español (Garmin las saca en inglés). Después, genera un diccionario de comentarios que indica cuáles de entre los aspectos de la rúbrica de sueño de Garmin tienen cada calificación, de tal manera que tengamos un insight final con la estructura: Excelente: aspecto x, aspecto, y, ..., Bueno: parámetro z, ..., etc.

El resto de la función sigue la misma estructura que el resto de las funciones de evaluación.

Listing C.4: 'Función de evaluación del sueño'

```
#Evaluacion del sueno
def evaluar_sueno(df):
    #Diccionario para traducir las calificaciones
    traducciones = {
        'EXCELLENT': 'Excelente',
        'GOOD': 'Bueno',
        'FAIR': 'Regular',
        'POOR': 'Malo'
    }

    resultados = []

    for index, row in df.iterrows():
        dia = row['Dia']
        calificacion_global = traducciones.get(row['Calificacion_Global'], row[
            'Calificacion_Global'])

        #Generar insights segun cada aspecto de la rebrica de Garmin
        aspectos = ['Duracion_Total', 'Estres', 'Inquietud', 'Despertar', 'REM',
                   'Sueno_Ligero', 'Sueno_Profundo']
        insights = {
            'Excelente': [],
            'Bueno': [],
            'Regular': [],
            'Malo': []
        }

        for aspecto in aspectos:
            calificacion = row[aspecto]
            calificacion_traducida = traducciones.get(calificacion,
                calificacion)

            if calificacion_traducida == 'Excelente':
                insights['Excelente'].append(aspecto)
            elif calificacion_traducida == 'Bueno':
                insights['Bueno'].append(aspecto)
```

## APÉNDICE C. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ALGORITMO DE CALIFICACIÓN EN PYTHON

```
        elif calificacion_traducida == 'Regular':
            insights['Regular'].append(aspecto)
        elif calificacion_traducida == 'Malo':
            insights['Malo'].append(aspecto)

    comentario = f"Aspectos del sueño el {dia}: "
    for calificacion, items in insights.items():
        if items:
            comentario += f"{calificacion}: {', '.join(items)}."

#Diccionario de resultados diarios
resultado_dia = {
    'Dia': dia,
    'Calificacion_Global': calificacion_global,
    'Insight_Sueno': comentario.strip()
}

resultados.append(resultado_dia)
#Devolvemos DF
return pd.DataFrame(resultados)

#Llamamos a la funcion
resultados_sueno = evaluar_sueno(df_sueno)
```

Listing C.4: 'Función de evaluación del sueño'

Listing C.5: 'Función de evaluación del balance de respiración pre/post entrenamiento'

```
#Evaluacion del balance de respiracion pre/post entrenamiento
df_resp_reducido = df_respiracion[['Dia', 'Balance_Respiracion']]

def evaluar_respiracion(df, insights_dict):
    resultados = []

    for index, row in df.iterrows():
        dia = row['Da']
        balance_resp = row['Balance_Respiracion']

        #Evaluacion
        if pd.notna(balance_resp):
            if balance_resp <= 0.5:
                calificacion_resp = 'Excelente'
            elif balance_resp <= 1:
                calificacion_resp = 'Buena'
            elif balance_resp <= 1.5:
                calificacion_resp = 'Regular'
            else:
                calificacion_resp = 'Mala'
        else:
            calificacion_resp = 'Mala'

        #Insight
```

## APÉNDICE C. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ALGORITMO DE CALIFICACIÓN EN PYTHON

```
comentario_resp = f"_{insights_dict[calificacion_resp]} ['Balance_U  
Respiracion'] }_{(Valor:_{balance_resp:.2f})_{rpm})}"  
  
#Diccionario de resultados diarios  
resultado_dia = {  
    'Da': dia,  
    'Calificacion_Balance_Respiracion': calificacion_resp,  
    'Insight_Balance_Respiracion': comentario_resp  
}  
resultados.append(resultado_dia)  
  
return pd.DataFrame(resultados)  
  
# Llamada a la funcion y visualizacin de resultados  
resultados_resp = evaluar_respiracion(df_resp_reducido, insights_dict_resp)
```

Listing C.5: 'Función de evaluación del balance de respiración pre/post entrenamiento'

La función de evaluación de estrés y body battery también tiene una estructura algo diferente.

En primer lugar, destaca el procesamiento de los DF iniciales y finales de cara a añadir la calificaciones de parámetros del 'día anterior' a las rúbricas. Para esto se usan las funciones *loc* y *shift* de *Pandas*, comentadas con anterioridad.

En cuanto a la extracción de los datos, se organiza de manera distinta ya que todos los parámetros de estrés se califican con el mismo criterio, por lo que se ordenan en una lista en la que se itera para usar una única estructura condicional de evaluación para dichos parámetros. Por tanto, definimos el diccionario de resultados diarios al inicio de la función, y se almacenan en él los resultados creando las columnas directamente después de calcularse.

Listing C.6: 'Función de evaluación de estrés y body battery'

```
#Evaluacion del BB y el estres  
  
#Pre procesamiento de DFs de BB y estres  
df_body_battery_reducido = df_body_battery[['Dia', 'Body_Battery_Max', 'Body_U  
Battery_Final', 'Body_Battery_Pre-Entrenamiento']]  
df_body_battery_reducido = df_body_battery_reducido.copy()  
df_body_battery_reducido['Body_Battery_Final_Dia_Anterior'] =  
    df_body_battery_reducido['Body_Battery_Final'].shift(1)  
df_body_battery_reducido.loc[[2,5], 'Body_Battery_Final_Dia_Anterior'] = None  
  
df_estres['Estres_Post_Entrenamiento_Dia_Anterior'] = df_estres['Estres_Post-  
Entrenamiento'].shift(1)  
df_estres.loc[[2,5], 'Estres_Post_Entrenamiento_Dia_Anterior'] = None  
df_estres_bb = pd.merge(df_estres, df_body_battery_reducido, on='Dia', how='  
outer')  
  
def evaluar_estres_body_battery(df, insights_dict_estres,  
    insights_dict_body_battery):  
    resultados = []  
  
    for index, row in df.iterrows():  
        dia = row['Dia']
```

## APÉNDICE C. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ALGORITMO DE CALIFICACIÓN EN PYTHON

```
resultado_dia = {'Dia': dia} #Declaramos el diccionario de manera inicial por la evaluacion distinta del estres.

#Evaluacion del estres. Todos los parametros de estres se evaluan con el mismo criterio.
parametros_estres = [
    'Estres\u201cPre-Entrenamiento', 'Estres\u201cPost-Entrenamiento',
    'Estres\u201cMedio\u201cDia\u201cDescanso', 'Estres\u201cPost\u201cEntrenamiento\u201cDia\u201cAnterior'
]

for param in parametros_estres:
    valor = row[param]
    if pd.notna(valor):
        if valor <= 25:
            calificacion = 'Excelente'
        elif valor <= 50:
            calificacion = 'Bueno'
        elif valor <= 75:
            calificacion = 'Regular'
        else:
            calificacion = 'Malo'

        #Creamos columnas de calificacion e insights directamente.
        resultado_dia[f'Calificacion\u201c{param}\u201d'] = calificacion
        resultado_dia[f'Insight\u201c{param}\u201d'] = f"{{insights_dict_estres[calificacion][param]}}\u201cValor:\u201c{valor}}"

#Evaluacion de parametros de body battery
valor_max = row['Body\u201cBattery\u201cMax']
if pd.notna(valor_max):
    if valor_max < 25:
        calificacion = 'Malo'
    elif valor_max < 50:
        calificacion = 'Regular'
    elif valor_max < 85:
        calificacion = 'Bueno'
    else:
        calificacion = 'Excelente'

    resultado_dia['Calificacion\u201cBody\u201cBattery\u201cMax\u201d'] = calificacion
    resultado_dia['Insight\u201cBody\u201cBattery\u201cMax\u201d'] = f"{{insights_dict_body_battery[calificacion]['Body\u201cBattery\u201cMax']}}\u201cValor:\u201c{valor_max}}"

valor_final = row['Body\u201cBattery\u201cFinal']
if pd.notna(valor_final):
    if valor_final < 10:
        calificacion = 'Malo'
    elif valor_final < 15:
        calificacion = 'Regular'
```

## APÉNDICE C. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ALGORITMO DE CALIFICACIÓN EN PYTHON

```
    elif valor_final < 25:
        calificacion = 'Bueno'
    else:
        calificacion = 'Excelente'

    resultado_dia['Calificacion_Body_Battery_Final'] = calificacion
    resultado_dia['Insight_Body_Battery_Final'] = f"{
        insights_dict_body_battery[calificacion]['Body_Battery_Final']}"]

    valor_pre = row['Body_Battery_Pre-Entrenamiento']
    if pd.notna(valor_pre):
        if valor_pre < 20:
            calificacion = 'Malo'
        elif valor_pre < 35:
            calificacion = 'Regular'
        elif valor_pre < 50:
            calificacion = 'Bueno'
        else:
            calificacion = 'Excelente'

        resultado_dia['Calificacion_Body_Battery_Pre-Entrenamiento'] =
            calificacion
        resultado_dia['Insight_Body_Battery_Pre-Entrenamiento'] = f"{
            insights_dict_body_battery[calificacion]['Body_Battery_Pre-
            Entrenamiento']}"]"(Valor:{valor_pre})"

    resultados.append(resultado_dia)

    return pd.DataFrame(resultados)

#Llamamos a la funcion
resultados_estres_bb = evaluar_estres_body_battery(df_estres_bb,
    insights_dict_estres, insights_dict_body_battery)

#Calificación de BB final dia anterior desplazando BB final
resultados_estres_bb = resultados_estres_bb.copy()
resultados_estres_bb['Calificacion_Body_Battery_Final_Dia_Anterior'] =
    resultados_estres_bb['Calificacion_Body_Battery_Final'].shift(1)
```

Listing C.6: 'Función de evaluación de estrés y body battery'

### C.0.2. Funciones de calificación

Listing C.7: Diccionarios con pesos para la calificación de la recuperación y el rendimiento

```
#Pesos para ponderacion de parametros de recuperacion por tipo de dia
pesos_descanso = { #dias de descanso
    'Calidad_del_Sueno': 2,
    'HRV': 2,
    'Body_Battery_Inicial': 1,
```

## APÉNDICE C. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ALGORITMO DE CALIFICACIÓN EN PYTHON

```
'Body_Battery_Final': 2,  
'Estres_Dia_Descanso': 1  
}  
  
pesos_entrenamiento_post_descanso = { #dias de entrenamiento tras dia de descanso  
    'Calificacion_Recuperacion_Dia_Descanso_Anterior': 2,  
    'Calidad_del_Sueno': 2,  
    'HRV': 2,  
    'Body_Battery_Inicial': 1,  
    'Body_Battery_Pre-Entrenamiento': 2,  
    'Estres_Pre-Entrenamiento': 1  
}  
  
pesos_entrenamiento_post_entrenamiento = { #dias de entrenamiento tras dia de entrenamiento  
    'Calidad_del_Sueno': 2,  
    'HRV': 2,  
    'Body_Battery_Inicial': 1,  
    'Body_Battery_Final_Dia_Anterior': 1,  
    'Body_Battery_Pre-Entrenamiento': 2,  
    'Estres_Pre-Entrenamiento': 1,  
    'Estres_Post_Entrenamiento_Dia_Anterior': 1  
}  
  
#Pesos para ponderacion de parametros de rendimiento  
pesos_parametros_rend = {  
    'Frecuencia_Cardiaca_Media': 3,  
    'Picos_Frecuencia_Cardiaca': 1,  
    'Valles_Frecuencia_Cardiaca': 1,  
    'Balance_Respiracion': 2  
}
```

Listing C.7: Diccionarios con pesos para la calificación de la recuperación y el rendimiento

Listing C.8: Función de calificación del rendimiento

```
#Funcion para evaluar el rendimiento del entrenamiento  
def calificar_rendimiento(resultados_fc, resultados_resp, resultados_vo2,  
    pesos_parametros):  
    resultados = []  
  
    for index in range(len(resultados_fc)):  
        dia = resultados_fc.loc[index, 'Dia']  
  
        #Declaramos valores nulos para los dias sin entrenamiento  
        if dia in ["Miercoles", "Sabado"]:  
            resultado_dia = {  
                'Dia': dia,  
                'Calificacion_Rendimiento': None,  
                'Insight_Rendimiento': None,  
                'Calificacion_V02_Maximo': None,
```

## APÉNDICE C. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ALGORITMO DE CALIFICACIÓN EN PYTHON

```
'Calificacion\uFrecuencia\uCardiaca\uMedia': None,
'Calificacion\uPicos\uFrecuencia\uCardiaca': None,
'Calificacion\uValles\uFrecuencia\uCardiaca': None,
'Calificacion\uBalance\uRespiracion': None
}
resultados.append(resultado_dia)
continue

#Extraccion de datos
parametros = {
    'Frecuencia\uCardiaca\uMedia': (resultados_fc.loc[index, 'Calificacion\uhr_medio'], resultados_fc.loc[index, 'Insight\uhr_medio']),
    'Picos\uFrecuencia\uCardiaca': (resultados_fc.loc[index, 'Calificacion\uPicos\uMedios'], resultados_fc.loc[index, 'Insight\uPicos\uMedios']),
    'Valles\uFrecuencia\uCardiaca': (resultados_fc.loc[index, 'Calificacion\uValles\uMedios'], resultados_fc.loc[index, 'Insight\uValles\uMedios']),
    'Balance\uRespiracion': (resultados_resp.loc[index, 'Calificacion\uBalance\uRespiracion'], resultados_resp.loc[index, 'Insight\uBalance\uRespiracion'])
}
calificacion_vo2 = resultados_vo2.loc[0, 'Calificacion\uV02']

## Sistema de Calificacion ##

#Calculo de la puntuacion

puntuacion_total = 0
total_pesos = 0
insights = []
calificaciones_individuales = {'Calificacion\uV02\uMaximo':
    calificacion_vo2} # Inicializamos el diccionario con V02 para desglose de la calificacion global

for param, (calificacion, comentario) in parametros.items():
    if pd.notna(calificacion):
        if calificacion == 'Excelente':
            puntos = 4
        elif calificacion == 'Bueno' or calificacion == 'Buena':
            puntos = 3
        elif calificacion == 'Regular':
            puntos = 2
        else:
            puntos = 1

        #Se pondra segun el peso del parametro
        peso = pesos_parametros.get(param, 1)
```

## APÉNDICE C. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ALGORITMO DE CALIFICACIÓN EN PYTHON

```
puntuacion_total += puntos * peso
total_pesos += peso

#Estructura final de los insights
insights.append(f'{param}: {calificacion}-{comentario}')

#Se añade la calificación a un diccionario para el desglose de
#la calificación global
calificaciones_individuales[f'Calificacion_{param}'] =
    calificacion

#Ajuste de los límites de puntuación según el VO2 máximo
if calificacion_vo2 == 'Excelente':
    limites = {'excelente': 3.5, 'bueno': 2.5, 'regular': 1.5} # Más
        restrictivo
elif calificacion_vo2 == 'Bueno':
    limites = {'excelente': 3.4, 'bueno': 2.4, 'regular': 1.4}
elif calificacion_vo2 == 'Regular':
    limites = {'excelente': 3.3, 'bueno': 2.3, 'regular': 1.3}
else:
    limites = {'excelente': 3.2, 'bueno': 1.2, 'regular': 1.2} # Menos
        permisivo

#Calificación según la puntuación

puntuacion_promedio = puntuacion_total / total_pesos
if puntuacion_promedio >= limites['excelente']:
    calificacion_global = 'Excelente'
elif puntuacion_promedio >= limites['bueno']:
    calificacion_global = 'Buena'
elif puntuacion_promedio >= limites['regular']:
    calificacion_global = 'Regular'
else:
    calificacion_global = 'Mala'

## Resultados ##

#Anadimos al diccionario
resultado_dia = {
    'Dia': dia,
    'Calificacion_Rendimiento': calificacion_global,
    **calificaciones_individuales,
    'Insight_Rendimiento': "\n".join(insights)
}
resultados.append(resultado_dia)

return pd.DataFrame(resultados)

#Se almacena el DF con los resultados llamando a la función
```

## APÉNDICE C. CÓDIGO COMPLEMENTARIO PARA LA IMPLEMENTACIÓN DEL ALGORITMO DE CALIFICACIÓN EN PYTHON

```
resultados_rendimiento = calificar_rendimiento(resultados_fc, resultados_resp,  
resultados_vo2, pesos_parametros_rend)
```

Listing C.8: Función de calificación del rendimiento



## D. Herramientas software utilizadas

En este apéndice se resume la lista completa de equipo software (herramientas, aplicaciones, librerías, programas, etc.) utilizada para el desarrollo completo del Trabajo, así como su distribuidor o desarrollador y el propósito de su uso dentro del ámbito del Proyecto.

Elemento Software	Distribuidor	Propósito
Sistema Operativo Windows 11	Windows	S.O. del PC con el que se ha desarrollado el Proyecto.
Anaconda Navigator	Anaconda	Interfaz gráfica de usuario que integra el entorno de Python de notebooks Jupyter
Jupyter notebooks	Jupyter Project	Entorno de desarrollo de Python.
Librería Pandas	NumFOCUS	Librería para el análisis de datos en Python.
Módulo os	Python Software Foundation	Módulo para la interacción con el SO en Python.
Módulo json	Python Software Foundation	Módulo para manipulación de archivos y datos JSON en Python.
Módulo re	Python Software Foundation	Módulo para emplear patrones con expresiones regulares en Python.
Módulo matplotlib.pyplot	NumFOCUS	Módulo para la representación gráfica de los datos en Python.
Módulo scipy.signal	NumFOCUS	Módulo para el procesado de señales.
Librería widgets	IPython	Librería para agregar elementos interactivos en Python.
Módulo display	IPython	Módulo para la visualización de salidas de formato complejo en Python.
Virtual Studio Code	Microsoft	Editor de código para el desarrollo de la app Garmin Connect IQ en Monkey C.
Extensión Connect IQ en VS Code	Garmin	Extensión que permite el desarrollo de aplicaciones Connect IQ en Monkey C.
Microsoft Word	Microsoft	Editor de texto en el que se confeccionaron las tareas individuales antes de la confección del informe.
Microsoft Excel	Microsoft	Editor de hojas de cálculo empleado para la el cálculo de la puntuación de la encuesta SUS.
Overleaf LaTeX	Overleaf	Editor de texto empleado para la confección de este informe.
Google Scholar	Google LLC	Motor de búsqueda enfocado en literatura académica, utilizado para la documentación del Proyecto.

*Continúa en la siguiente página*

## APÉNDICE D. HERRAMIENTAS SOFTWARE UTILIZADAS

---

Elemento Software	Distribuidor	Propósito
Statista	Statista, Inc	Plataforma estadística de mercado utilizada para la documentación del Proyecto.
Zotero	Corporation for Digital Scholarship	Herramienta para la gestión de las referencias bibliográficas que fundamentan el Proyecto.
Canva	Canva Pty Ltd	Herramienta de diseño gráfico, utilizada para la confección de gráficos para el informe y para la presentación de la defensa del Proyecto.
Lucidchart	Lucid Software Inc	Herramienta para la creación de diagramas de flujo, empleada para el diagrama de flujo de la implementación del desarrollo técnico.

Cuadro D.1: Elementos software utilizados en el desarrollo del Proyecto

---

# Listado de acrónimos

**AA** Asalto de Actividad. 36, 38

**AR** Asalto de Recuperación. 36, 38

**ASR** Arritmia Sinusal Respiratoria. 45

**BB** Body Battery. 41, 62, 63

**BMR** Tasa Metabólica Basal. 44

**CMB** Consejo Mundial de Boxeo. 35, 42

**DA** Día Anterior. 61, 62, 64

**DF** DataFrame. 57, 82

**FC** Frecuencia Cardíaca. 39, 42, 65, 126

**FEK** Federación Española de Kickboxing. 35

**FR** Frecuencia Respiratoria. 42, 65, 119

**HRR** Frecuencia Cardíaca de Reserva. 39, 66

**HRV** Variabilidad de la Frecuencia Cardíaca. 39, 41, 43, 62, 72, 149

**LOPDGDD** Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales. 107

**SNA** Sistema Nervioso Autónomo. 41, 45

**ZFCO** Zona de Frecuencia Cardíaca Óptima. 39, 65, 66



