



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



TFG del Grado en Ingeniería  
Informática

título del TFG  
Documentación Técnica



Presentado por José Luis Pérez Gómez  
en Universidad de Burgos — 30 de junio  
de 2024

Tutor: Bruno Baruque



---

# Índice general

---

<b>Índice general</b>	<b>i</b>
<b>Índice de figuras</b>	<b>iii</b>
<b>Índice de tablas</b>	<b>iv</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	2
A.3. Estudio de viabilidad . . . . .	7
<b>Apéndice B Especificación de Requisitos</b>	<b>9</b>
B.1. Introducción . . . . .	9
B.2. Objetivos generales . . . . .	9
B.3. Catálogo de requisitos . . . . .	10
B.4. Especificación de requisitos . . . . .	13
<b>Apéndice C Especificación de diseño</b>	<b>15</b>
C.1. Introducción . . . . .	15
C.2. Diseño de datos . . . . .	15
C.3. Diseño procedimental . . . . .	15
C.4. Diseño arquitectónico . . . . .	15
<b>Apéndice D Documentación técnica de programación</b>	<b>17</b>
D.1. Introducción . . . . .	17
D.2. Estructura de directorios . . . . .	17
D.3. Manual del programador . . . . .	17

D.4. Compilación, instalación y ejecución del proyecto . . . . .	17
D.5. Pruebas del sistema . . . . .	17
<b>Apéndice E Documentación de usuario</b>	<b>19</b>
E.1. Introducción . . . . .	19
E.2. Requisitos de usuarios . . . . .	19
E.3. Instalación . . . . .	19
E.4. Manual del usuario . . . . .	19
<b>Apéndice F Anexo de sostenibilización curricular</b>	<b>21</b>
F.1. Introducción . . . . .	21

---

## Índice de figuras

---

A.1. Commits realizados durante la realización del TFG . . . . .	2
A.2. Additions realizados durante la realización del TFG . . . . .	3
A.3. Deletions realizados durante la realización del TFG . . . . .	3
A.4. Commits-Etapa 1 . . . . .	4
A.5. Commits-Etapa 2 . . . . .	5
A.6. Commits-Etapa 3 . . . . .	6

---

# Índice de tablas

---

B.1. CU-1 Nombre del caso de uso. . . . .	14
---	----

## *Apéndice A*

---

# **Plan de Proyecto Software**

---

### **A.1. Introducción**

La Universidad de Burgos, dentro del área de conocimiento de Ingeniería de Sistemas y Automática, dispone de un interfaz BCI (Brain Computer Interface) para la captación de señales cerebrales. Empleando ese interfaz se han realizado diferentes experimentos que han permitido recoger información de la actividad cerebral mientras los usuarios ejecutaban diferentes tareas cotidianas.

Este Trabajo de Fin de Grado (TFG) tiene como objetivo el análisis de la información obtenida en esos experimentos. Se entrenarán diferentes algoritmos para clasificar la acción realizada por el usuario a partir de las señales generadas por el BCI. Con este propósito, se evaluarán diferentes algoritmos de procesamiento de señales y de machine/deep learning para la clasificación automática de señales.

Los datos aportados son de tipo EEG (Electroencefalografía) para la realización del TFG son datos referentes a experimentos basados en pulsaciones sobre teclas de un teclado: arriba, abajo, izquierda, derecha.

El análisis de estos datos y su evaluación en diferentes algoritmos esta basada en predecir qué teclas del teclado se han pulsado según las señales captadas con la interfaz BCI.

Para esto no ha habido una planificación como tal registrada en Github, pero sí una progresión definida en los commits de código generados durante la composición del TFG.

## A.2. Planificación temporal

En la reunión inicial con los tutores definimos utilizar Python para realizar el código y una serie de aprendizajes básicos para poder acometer el TFG sin problemas.

En las siguientes reuniones se definieron algoritmos y experimentos a realizar con los datos EGG aportados.

Desde el principio del proyecto debido a las circunstancias personales del alumno no se ha podido realizar metodología Scrum, pero sí se produjeron estas líneas temporales:

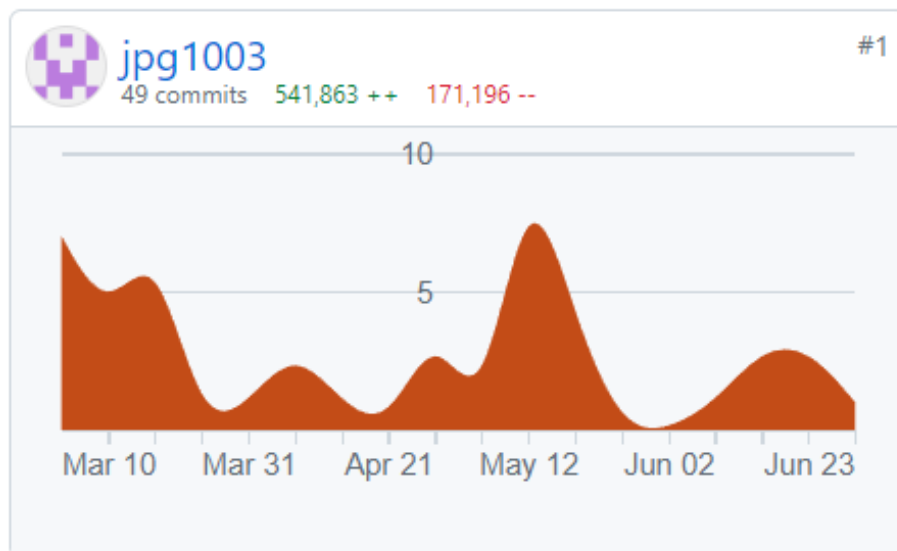


Figura A.1: Commits realizados durante la realización del TFG



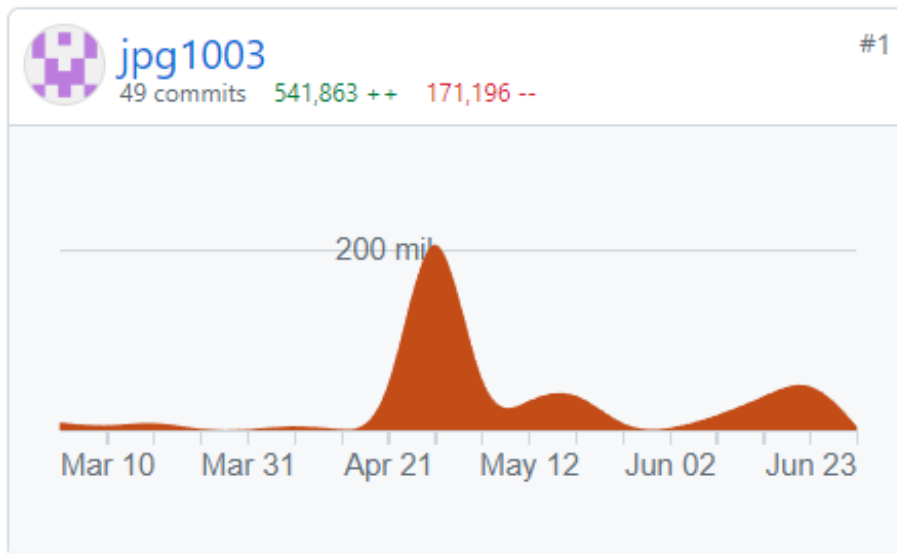


Figura A.2: Additions realizados durante la realización del TFG



Figura A.3: Deletions realizados durante la realización del TFG

Se pueden dividir en 3 grandes etapas:

- Etapa de estudio: (Febrero a Marzo)
- Etapa de desarrollo: (Abril a Mayo)
- Etapa de desarrollo y optimización: (Junio a Julio)

## Etapa de estudio: (Febrero a Marzo)

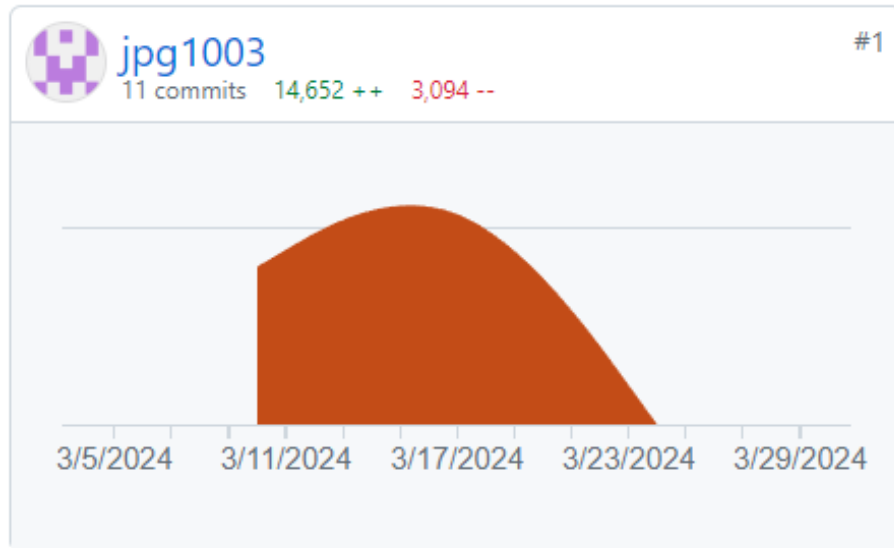


Figura A.4: Commits-Etapa 1

- **Realización de cursos online** propuestos por los tutores Bruno Baruque y Jesús Enrique Sierra. [Tutorial de Inicio de Pandas](#), [Visualización de Datos](#), [Trabajo con series temporales](#)

- **Análisis del conjunto de datos**, en archivo csv, proporcionado por los tutores.
- **Creación esqueleto para la estructura el TFG en Github.**
- **Definición y creación de los primeros notebooks**, principalmente análisis del conjunto de datos.
- **Definición y creación de los primeros notebooks de machine learning**, al final de la etapa.
- **Comentar código e imprimir comentarios en los notebooks.**

Los principales problemas o obstáculos que me encontré fueron principalmente los siguientes:

- **Tiempo invertido en la realización de los cursos.**
- **Estructuración del código.** Me tomo mucho tiempo poder llegar a definir como quería mostrar el código, me decidí por un notebook principal que realizara llamadas al resto de notebooks con códigos mas específicos para cada modelo o experimento a realizar.
- **Plotteos y definiciones básicas como normalizar o escalar el conjunto de datos.**

## Etapa de desarrollo: (Abril a Mayo)

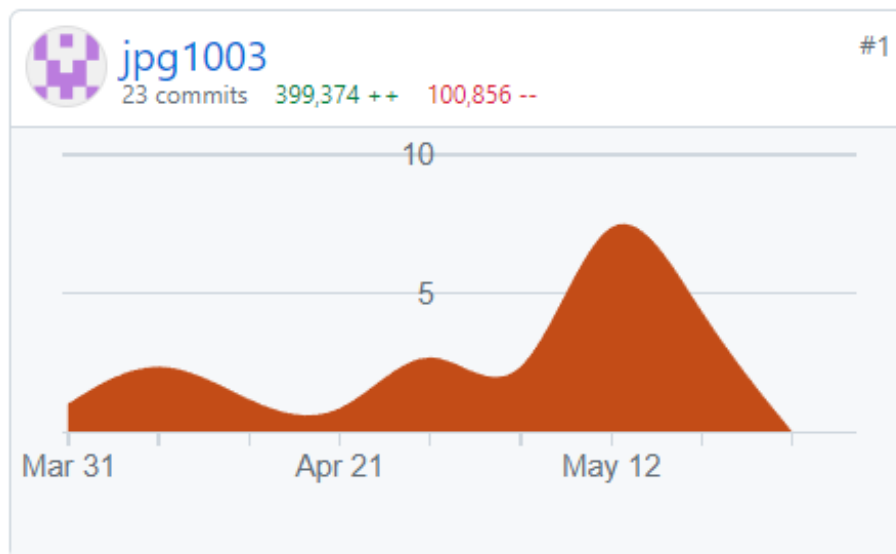


Figura A.5: Commits-Etapa 2

- Cambio de análisis del conjunto de datos, preprocessing.
- Definición y creación notebooks de machine learning.
- Definición y creación notebooks de deep learning.

Los problemas que me encontré en esta etapa fueron:

- **Compilación modelos deep learning.** Al ejecutar los primeros modelos de deep learning los datos normalizados me proporcionaban errores de compilación con modelos SRNN o LSTM, cambiando a datos escalados y shapeando los modelos pudieron ejecutarse.

- **Utilización de ventanas temporales en los modelos deep learning.** En esta etapa no supe identificar este requerimiento por parte de los tutores y estuve implementando varias formas de poder utilizar ventanas temporales en el código.

- Utilización de modelos deep learning y callbacks.

- Gráficas y definiciones básicas como normalizar o escalar el conjunto de datos.

**Etapas de desarrollo y optimización: (Junio a Julio)**

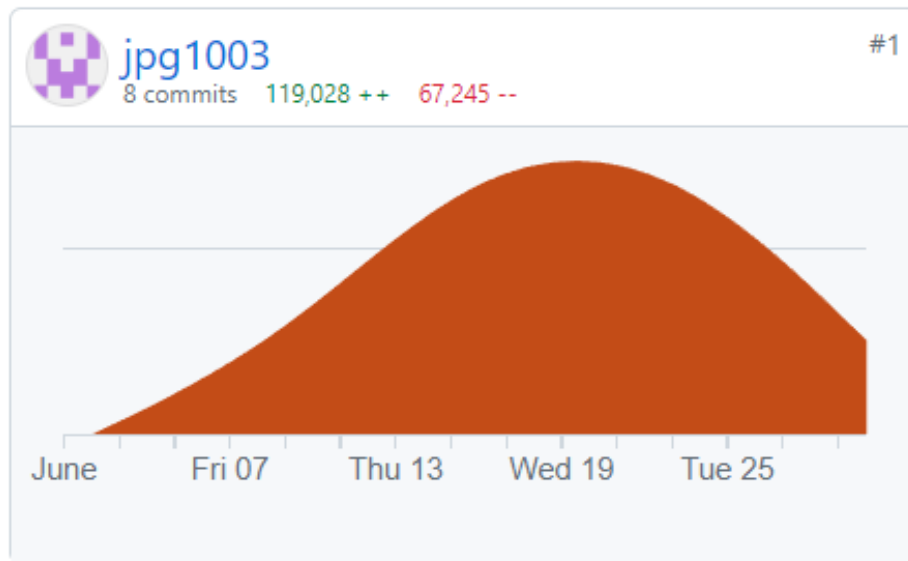


Figura A.6: Commits-Etapa 3

- **Añado nuevos gráficos** en análisis del conjunto de datos, preprocesing.
- **Definición y creación ventanas temporales acordadas con Bruno Baruque.**
- **Definición y creación nuevos notebooks de deep learning**
- **Definición y creación nuevos datos sintéticos a través de el aplicativo smote.**
- **Continuar con el comentado del código e imprimir comentarios en los notebooks.**

En la última etapa los problemas que se han acontecido son:

- **Utilización de ventanas temporales en los modelos deep learning.** Después de varias algunas reuniones con Bruno Baruque se llego a la defincion correcta para las ventanas temporal en el conjunto de datos.
- **Utilización de modelos deep learning y callbacks.**
- **Utilización datos sintéticos.** Definir correctamente esta generacion de datos y poder utilizarlos en los modelos deep learning correctamente.

## **A.3. Estudio de viabilidad**

**Viabilidad económica**

**Viabilidad legal**



## *Apéndice B*

---

# Especificación de Requisitos

---

## B.1. Introducción

La Especificación de Requisitos tiene como objetivo definir de manera clara y detallada las necesidades y expectativas del proyecto que consta de un notebook Jupyter definido en Python, el cual invoca a otros notebooks Jupyter secundarios para realizar diversas tareas.

Este documento sirve para asegurar que todas las partes interesadas comprendan y acuerden los objetivos y funcionalidades del sistema a desarrollar.

En un entorno de trabajo basado en notebooks Jupyter, es esencial contar con una especificación precisa que guíe el desarrollo, implementación y validación del sistema. Esto no solo facilita la colaboración y la comunicación entre los desarrolladores, analistas de datos y usuarios finales, sino que también ayuda a identificar posibles riesgos y desafíos durante el ciclo de vida del proyecto.

## B.2. Objetivos generales

Los objetivos principales de este TFG son los siguientes:

- **Facilitar el procesamiento de datos EEG.** Generados en la universidad mediante el interfaz BCI.
- **Automatizar tareas repetitivas.** Con la creación de notebooks secundarios para poder ejecutar las tareas comunes y repetitivas relacionadas con los datos EEG.

- **Mejorar la tasa de acierto en los analisis de modelos** implementando modelos precisos y fiables.
- **Asegurar la accesibilidad y usabilidad en los notebooks.** Los notebooks son herramientas intuitivas y accesibles que aportan esa facilidad a la hora de interactuar.
- **Integracion y extensibilidad.** Al utilizar llamadas a otros notebooks se asegura que se permita la integracion de nuevos notebooks y se asegura posibles nuevas mejoras en el codigo.

### B.3. Catálogo de requisitos

Hay dos tipos de requisitos, los funcionales (qué debe hacer el código) y los no funcionales (cómo debe funcionar el código):

#### Requisitos funcionales

- **RF-001 Iniciar notebook Main:**
  - **Descripción:** Los usuarios han de poder iniciar el notebook principal para poder comenzar con la ejecución del trabajo.
  - **Prioridad:** Alta
  - **Criterios de aceptación:** El notebook Main debe abrirse correctamente en menos de 5 segundos.
- **RF-002 Invocación de Notebooks Secundarios Preparatorios:**
  - **Descripción:** Los usuarios han de poder invocar los notebook secundarios preparatorios para poder ejecutar con seguridad el resto de notebooks de preprocesado y modelados.
  - **Prioridad:** Alta
  - **Criterios de aceptación:** Los notebooks secundarios preparatorios se han de ejecutar correctamente y que la celda en la que se llama al notebook secundario no devuelva errores. Los notebooks preparatorios. Su ejecución no ha de ser mayor a 5 minutos.
- **RF-003 Invocación de Notebook Secundarios carga de archivo CSV:**
  - **Descripción:** Los usuarios han de poder invocar los notebook secundarios carga de archivo CSV, para la ejecución de este notebook secundario se ha de subir un archivo csv con el conjunto



de datos a analizar y una vez subido el archivo, se iluminara un botón con nombre 'Procesar datos' que procesara los datos una vez pulsado.

- **Prioridad:** Alta
  - **Criterios de aceptación:** Si no se ejecuta correctamente la llamada a este notebook no se podrá seguir con la ejecución del resto de notebooks puesto que sin datos no se podrían realizar las siguientes acciones.
- **RF-004 Invocación de Notebooks Secundarios análisis y pre-processing:**
- **Descripción:** Los usuarios han de poder invocar los notebook secundarios análisis y preprocessing para poder proporcionar este procesamiento de datos antes de la ejecución de modelos machine o deep learning.
  - **Prioridad:** Alta
  - **Criterios de aceptación:** Si no se ejecuta correctamente la llamada a este notebook no se podrá seguir con la ejecución del resto de notebooks puesto que sin la realización del preprocessing de los datos no se generan los csv correspondientes para poder seguir ejecutando los siguientes notebooks.
- **RF-005 Invocación de Notebooks Secundarios para experimentos:**
- **Descripción:** Los usuarios han de poder invocar los notebook secundarios para experimentos y poder generar los datos sobre Tasa de acierto y generación de matrices de confusión.
  - **Prioridad:** Media
  - **Criterios de aceptación:** Estos notebooks se puede ejecutar de manera individual o en el orden pre establecido. para realizar el análisis final lo ideal es ejecutar todos los notebooks de esta session.
- **RF-006 Invocación de Notebook Secundario resultados:**
- **Descripción:** Los usuarios han de poder invocar los notebook Secundario resultados y poder visualizar los resultados de las particiones test del conjunto de datos. Se han de haber ejecutado todos los notebooks de esta sección.
  - **Prioridad:** Alta
  - **Criterios de aceptación:** Sin la ejecución de este notebook no se obtendrían los resultados del trabajo y por lo tanto es critico

este ultimo notebook para poder visualizar la tasa de acierto en datos que el modelo no conocía en la etapa de entrenamiento y validación.

## Requisitos no funcionales

### ■ RNF-001 Rendimiento:

- **Descripción:** El sistema donde se ejecuten los notebooks ha de poder manejar la ejecución sin causar demoras significativas.
- **Criterios de aceptación:** El tiempo de ejecución para la ejecución de cada notebook secundario no ha de ser superior a 15 minutos.

### ■ RNF-002 Usabilidad:

- **Descripción:** Los notebooks han de ser faciles de entender y usar por los usuarios.
- **Criterios de aceptación:** La interfaz de los notebooks ha de ser intuitiva y clara.

### ■ RNF-003 Escalabilidad:

- **Descripción:** En la configuración del notebooks principal se debe permitir la adhesión de nuevos notebooks y sus llamadas desde el notebook principal.
- 
- **Criterios de aceptación:** Poder seguir integrando nuevos notebooks para ser llamados desde el notebook principal sin afectar al funcionamiento del código.

## B.4. Especificación de requisitos

CU-1	Ejemplo de caso de uso
<b>Versión</b>	1.0
<b>Autor</b>	José Luis Pérez Gómez
<b>Requisitos asociados</b>	RF-xx, RF-xx
<b>Descripción</b>	La descripción del CU
<b>Precondición</b>	Precondiciones (podría haber más de una)
<b>Acciones</b>	<ol style="list-style-type: none"><li>1. Pasos del CU</li><li>2. Pasos del CU (añadir tantos como sean necesarios)</li></ol>
<b>Postcondición</b>	Postcondiciones (podría haber más de una)
<b>Excepciones</b>	Excepciones
<b>Importancia</b>	Alta o Media o Baja...

Tabla B.1: CU-1 Nombre del caso de uso.



## *Apéndice C*

---

# **Especificación de diseño**

---

- C.1. Introducción
- C.2. Diseño de datos
- C.3. Diseño procedimental
- C.4. Diseño arquitectónico



## *Apéndice D*

---

# **Documentación técnica de programación**

---

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución del proyecto
- D.5. Pruebas del sistema





## *Apéndice E*

---

# **Documentación de usuario**

---

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario



## *Apéndice F*

---

# **Anexo de sostenibilización curricular**

---

### **F.1. Introducción**

Este anexo incluirá una reflexión personal del alumnado sobre los aspectos de la sostenibilidad que se abordan en el trabajo. Se pueden incluir tantas subsecciones como sean necesarias con la intención de explicar las competencias de sostenibilidad adquiridas durante el alumnado y aplicadas al Trabajo de Fin de Grado.

Más información en el documento de la CRUE [https://www.crue.org/wp-content/uploads/2020/02/Directrices\\_Sostenibilidad\\_Crue2012.pdf](https://www.crue.org/wp-content/uploads/2020/02/Directrices_Sostenibilidad_Crue2012.pdf).

Este anexo tendrá una extensión comprendida entre 600 y 800 palabras.