



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Aplicación de minería de  
datos a la detección de  
ataques epilépticos**



Presentado por José Luis Garrido Labrador  
en Universidad de Burgos — 16 de mayo  
de 2019

Tutores: Álgvar Arnáiz González y José  
Francisco Díez Pastor







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



Dr. D. Álgvar Arnáiz González, profesor del departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. José Luis Garrido Labrador, con DNI **TODO**, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Aplicación de minería de datos a la detección de ataques epilécticos.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 16 de mayo de 2019

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

Dr. D. Álgvar Arnáiz González

Dr. D. José Francisco Díez Pastor





## Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

## Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

## **Abstract**

A **brief** presentation of the topic addressed in the project.

## **Keywords**

keywords separated by commas.



---

# Índice general

---

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
Objetivos del proyecto	3
Conceptos teóricos	5
3.1. Secciones . . . . .	5
3.2. Referencias . . . . .	5
3.3. Imágenes . . . . .	6
3.4. Listas de ítems . . . . .	6
3.5. Tablas . . . . .	7
Técnicas y herramientas	9
4.1. Investigación . . . . .	9
4.2. Servicio web . . . . .	10
4.3. Herramientas generales . . . . .	11
Aspectos relevantes del desarrollo del proyecto	13
Trabajos relacionados	15
Conclusiones y Líneas de trabajo futuras	17

**Bibliografía**

**19**

---

# Índice de figuras

---

3.1. Autómata para una expresión vacía . . . . .	6
--	---

---

# Índice de tablas

---

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	8
---	---

---

# Introducción

---

Gracias al avance de las técnicas y algoritmos de minería de datos, disciplinas no directamente relacionadas con la computación se han ido beneficiando de las ciencias de datos, a modo particular, la medicina está desarrollándose hacia modelos más preventivos gracias a las predicciones que se pueden generar utilizando estos métodos.

Es por este motivo que podemos ver en la literatura científica de los últimos años como se relaciona medicina y ciencia de datos, por ejemplo en estudios de detección de caídas [23] o la motorización del sueño para la prevención de apneas [11]. En este trabajo de fin de grado, el objetivo es la detección de crisis epilépticas adscrito al proyecto homónimo vencedor del concurso universitario *Desafío Universidad Empresa* [4]

Aunque el análisis y detección de crisis epilépticas de manera automática ha sido ampliamente explorada por la comunidad científica esta se ha centrado o en el uso de pulseras inteligentes [17] o el uso de encefalogramas (*EEG*) [9, 12, 24] para la predicción de estos eventos. Por tanto, ante pacientes con diversos problemas de salud que impiden el uso de pulseras y de dispositivos de control de actividad cerebral se realiza este proyecto que enfoca la detección de las crisis epilépticas en el uso de sensores de presión y biométricos en la cama donde duerme el paciente.



---

## Objetivos del proyecto

---

El objetivo que se plantea es la creación de un clasificador que detecte cuando un paciente dormido padece de una crisis epiléptica, la interfaz de recopilación de datos y la interfaz por la cual el personal médico encargado de la salud del paciente conozca la situación de crisis.





---

# Conceptos teóricos

---

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de L<sup>A</sup>T<sub>E</sub>X<sup>1</sup>.

## 3.1. Secciones

Las secciones se incluyen con el comando `section`.

### Subsecciones

Además de secciones tenemos subsecciones.

### Subsubsecciones

Y subsecciones.

## 3.2. Referencias

Las referencias se incluyen en el texto usando `cite [?]`. Para citar webs, artículos o libros `[?]`.

---

<sup>1</sup>Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

### 3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de  $\text{\LaTeX}$ , pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:



Figura 3.1: Autómata para una expresión vacía

### 3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

**Primer item** más información sobre el primer item.

**Segundo item** más información sobre el segundo item.

▪

### 3.5. Tablas

Igualmente se pueden usar los comandos específicos de  $\text{\LaTeX}$  o bien usar alguno de los comandos de la plantilla.

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikTeX					X
TeXMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

---

# Técnicas y herramientas

---

## 4.1. Investigación

El proyecto ha sido desarrollado aplicando diversas técnicas de la minería de datos siguiendo el proceso KDD [19]. Aunque en esta memoria se explicarán superficialmente el desarrollo completo de la investigación se encuentra en el cuaderno de investigación adjuntado.

Para el proceso del clasificador final el flujo KDD ha sido el siguiente:

1. **Selección** Los datos han sido cedidos por la asociación abulense *PRONISA* que contiene jornadas nocturnas con identificadores de la cama, datos de presiones y datos vitales. Estos datos no están balanceados ya que existe una mayor cantidad de datos del paciente durmiendo que bajo una crisis epiléptica.
2. **Preprocesado** Los datos se han limpiado reduciendo ruidos mediante filtros de señal.
3. **Transformación** Se han generado datos estadísticos para series temporales que optimizaban el valor del área bajo la curva PCR [22] desde los datos preprocesados.
4. **Minería de datos** Se ha aplicado un sistema de clasificación doble, en primer lugar se utiliza un árbol de clasificación muy simple que divide entre las situaciones de despierto (bajas presiones en la cama) y acostado, en esta situación se aplica un clasificador *Random Forest* [3] para determinar si hay crisis o no.

5. **Interpretación** Se ha desplegado una aplicación web con datos en tiempo real para evaluar de manera constante la situación actual del paciente.

Este proceso de investigación se realizó sobre *Python* utilizando el ecosistema de librerías *SciPy* [10], en particular la librería de aprendizaje automático *scikit-learn* [15], de computación *NumPy* [14], de análisis de datos *Pandas* [13] y la de dibujado *Matplotlib* [8]. Además, en algunas partes de la investigación se utilizó *Weka* [7] para estudiar métodos que no se encontraban en *scikit-learn* como el *Random Balance* [5] o *Rotation Forest* [18].

## 4.2. Servicio web

### *Backend*

Las herramientas utilizadas para programar el servidor han sido:

**Flask** [21] *Microframework* de código abierto (BSD) que ofrece una capa de abstracción muy alta de un servicio web.

**Jinja** [20] Gestor de plantillas de código abierto (BSD) para Python.

**Flask-Socketio** [6] Integración del servicio de *sockets*, *Socket.IO* [1], compatible con los *WebSockets*.

**Gevent y Eventlet** [16, 2] Librerías para el uso de tiempo real asíncrono de hilos para el uso de *Socket.IO*.

Para la programación del sistema de hilos que distribuyen datos en tiempo real se siguieron los paradigmas de la programación orientada a objetos y de programación funcional. El sistema de rutas siguió las guías del *microframework Flask*.

Algunos patrones de diseño utilizados han sido el *Singleton* para la API.

### *Frontend*

Para el desarrollo de la parte visible de la aplicación se han usado otra serie de herramientas:

**Bootstrap** [26] *Framework* de *CSS* de código abierto (MIT) creado por *Twitter* para la creación de aplicaciones web redimensionables.

**jQuery** [27] *Framework* de *JavaScript* de código abierto (MIT) que simplifica el acceso al *HTML DOM* de la página.

**Chart.js** Librería de *JavaScript* de código abierto (MIT) para la creación de grafos en *canvas*.

## 4.3. Herramientas generales

Para el desarrollo general del proyecto se han utilizado las siguientes herramientas según el ámbito al que pertenecen:

### Servidor

**Nginx** servidor web y de proxy reverso ligero de alto rendimiento [29] de código abierto (BSD simplificada).

**MariaDB** sistema de gestión de bases de datos derivado de *MySQL* [28] de código abierto (GPLv2). Este motor es extremadamente compatible con *MySQL* porque es creado como una bifurcación de esto para garantizar la existencia de este motor bajo GPL.

**Proxmox** es un entorno de virtualización de servidores [30] de código abierto (AGPL). Su función principal es el despliegue y gestión de máquinas virtuales y contenedores.

**Anarchy Arch** sistema GNU/Linux derivado de *ArchLinux* [25] sobre el cual se ejecuta todo el servidor, está alojado en una máquina virtual del entorno *Proxmox*.

### Miscelanea

- **Jupyter Notebooks**: IDE de programación de *Python* basado en *iPython* de código abierto (BSD).
- **PyCharm Professional**: IDE de programación para *Python* avanzado basado en *IntelliJ*.

- **Visual Studio Code**: Editor de código genérico de código abierto (MIT).
- **Postman**: IDE para la ejecución de request *HTTP*.
- **Selenium**: IDE de pruebas sobre web de código abierto (APACHE).
- **CertBot**: Sistema para la firma *SSL* sobre *HTTP* gratuito de *LetsEncrypt* de código abierto (MPL).
- **Overleaf**: Editor de  $\text{\LaTeX}$ online para el trabajo colaborativo.
- **TexStudio**: Editor de  $\text{\LaTeX}$ de código abierto (GPLv2).
- **Dia**: Editor de diagramas genérico de código abierto (GPL).
- **StartUML**: Editor de diagramas UML.
- **Codesketch DB**: Traductor bidireccional de código-diagrama para bases de datos.
- **Filezilla**: Aplicación para la transferencia de ficheros sobre *FTP* y *SFTP* de código abierto (GPLv2).
- **GitHub**: Servicio online de *hosting* para repositorios Git.
- **ZenHub**: Servicio online de integración de herramientas *SCRUM* sobre GitHub.



---

## Aspectos relevantes del desarrollo del proyecto

---

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros<sup>3</sup>, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.



---

## Trabajos relacionados

---

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.



---

## **Conclusiones y Líneas de trabajo futuras**

---

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.



---

## Bibliografía

---

- [1] Damien Arrachequesne and Erik Little. Socket.io. <https://socket.io/docs/>, feb 2018.
- [2] Karl J Aström. Event based control. In *Analysis and design of nonlinear control systems*, pages 127–147. Springer, 2008.
- [3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [4] Radio Amiga Burgos. Álar Arnaiz González y José Francisco Díez Pastor – Premiadados del Concurso Desafío Universidad Empresa, Marzo 2018.
- [5] José F Díez-Pastor, Juan J Rodríguez, César García-Osorio, and Ludmila I Kuncheva. Random balance: ensembles of variable priors classifiers for imbalanced data. *Knowledge-Based Systems*, 85:96–111, 2015.
- [6] Miguel Grinberg. Flask-socketio. <https://flask-socketio.readthedocs.io/>, jan 2019.
- [7] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009.
- [8] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.
- [9] Jesper Jeppesen, Sándor Beniczky, A Fuglsang Frederiksen, Per Sidenius, and Peter Johansen. Modified automatic r-peak detection algorithm for patients with epilepsy using a portable electrocardiogram recorder. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4082–4085. IEEE, 2017.

- [10] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 07/02/2019].
- [11] Juha M Kortelainen, Mark Van Gils, and Juha Pärkkä. Multichannel bed pressure sensor for sleep monitoring. In *2012 Computing in Cardiology*, pages 313–316. IEEE, 2012.
- [12] Yatindra Kumar, ML Dewal, and RS Anand. Epileptic seizures detection in eeg using dwt-based apen and artificial neural network. *Signal, Image and Video Processing*, 8(7):1323–1334, 2014.
- [13] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.
- [14] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.
- [15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [16] Daniel Pope. Gevent: asynchronous i/o made easy. EuroPython, 2014. <https://doi.org/10.5446/19958> *Lastaccessed* : 16May2019.
- [17] Sriram Ramgopal, Sigride Thome-Souza, Michele Jackson, Navah Ester Kadish, Iván Sánchez Fernández, Jacquelyn Klehm, William Bosl, Claus Reinsberger, Steven Schachter, and Tobias Loddenkemper. Seizure detection, seizure prediction, and closed-loop warning systems in epilepsy. *Epilepsy & behavior*, 37:291–307, 2014.
- [18] Juan José Rodríguez, Ludmila I Kuncheva, and Carlos J Alonso. Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1619–1630, 2006.
- [19] Juan José Rodríguez. Introducción a la minería de datos.
- [20] Armin Ronacher, David Lord, Adrian Mönich, and Markus Unterwaditzer. Welcome to jinja2. <http://jinja.pocoo.org/docs/2.10/>, 2008.



- [21] Armin Ronacher, David Lord, Adrian Mönich, and Markus Unterwaditzer. Welcome to flask. <http://flask.pocoo.org/docs/1.0/>, 2010.
- [22] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.
- [23] Marie Tolkiehn, Louis Atallah, Benny Lo, and Guang-Zhong Yang. Direction sensitive fall detection using a triaxial accelerometer and a barometric pressure sensor. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 369–372. IEEE, 2011.
- [24] Alexandros T Tzallas, Markos G Tsipouras, Dimitrios G Tsalikakis, Evangelos C Karvounis, Loukas Astrakas, Spiros Konitsiotis, and Margaret Tzaphlidou. Automated epileptic seizure detection methods: a review study. In *Epilepsy-histological, electroencephalographic and psychological aspects*. IntechOpen, 2012.
- [25] Wikipedia contributors. Arch linux — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Arch\\_Linux&oldid=896735398](https://en.wikipedia.org/w/index.php?title=Arch_Linux&oldid=896735398), 2019. [Online; accessed 16-May-2019].
- [26] Wikipedia contributors. Bootstrap (front-end framework) — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Bootstrap\\_\(front-end\\_framework\)&oldid=895052706](https://en.wikipedia.org/w/index.php?title=Bootstrap_(front-end_framework)&oldid=895052706), 2019. [Online; accessed 16-May-2019].
- [27] Wikipedia contributors. JQuery — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=JQuery&oldid=896325154>, 2019. [Online; accessed 16-May-2019].
- [28] Wikipedia contributors. Mariadb — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=MariaDB&oldid=897072367>, 2019. [Online; accessed 16-May-2019].
- [29] Wikipedia contributors. Nginx — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Nginx&oldid=896866646>, 2019. [Online; accessed 16-May-2019].
- [30] Wikipedia contributors. Proxmox virtual environment — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Proxmox\\_Virtual\\_Environment&oldid=896960348](https://en.wikipedia.org/w/index.php?title=Proxmox_Virtual_Environment&oldid=896960348), 2019. [Online; accessed 16-May-2019].