

# MODELADO DE DATOS Y APLICACIÓN WEB PARA EL ESTUDIO DE LA ENFERMEDAD ARTERIAL PERIFÉRICA

Jorge Simón Gil



15 de diciembre de 2017

# Contents

## 1. Introducción

## 2. Herramientas

## 3. Resultados

## 4. Conclusiones

## 5. Líneas Futuras

# Introducción: Contexto de estudio

- ▶ La Enfermedad Arterial Periférica o EAP es una enfermedad que afecta al sistema cardiovascular obstruyendo tanto el riego como el flujo sanguíneo de las extremidades inferiores. Una de sus consecuencias es que durante la actividad física los pacientes padecen dolor en las zonas afectadas y la necesidad de detenerse, esto se conoce como claudicación intermitente, y cuando hablamos de actividad física nos referimos incluso caminar.

# Introducción: Contexto de estudio

- ▶ La Enfermedad Arterial Periférica o EAP es una enfermedad que afecta al sistema cardiovascular obstruyendo tanto el riego como el flujo sanguíneo de las extremidades inferiores. Una de sus consecuencias es que durante la actividad física los pacientes padecen dolor en las zonas afectadas y la necesidad de detenerse, esto se conoce como claudicación intermitente, y cuando hablamos de actividad física nos referimos incluso caminar.
- ▶ Además los grupos de riesgo del estudio de la EAP forman un gran porcentaje de la población mundial
  - ▶ Diabéticos: 1 de cada 11 personas.
  - ▶ Enfermedades Cardiovasculares: 31 % de la población.
  - ▶ Enfermedades Respiratorias: 235 millones de los 7000 millones de habitantes en el mundo.

# Introducción: Objetivo

- ▶ ¿Cómo podemos conseguir datos de calidad para el estudio a bajo coste?, ¿Qué es indispensable para la realización posterior de conclusiones?

# Introducción: Objetivo

- ▶ ¿Cómo podemos conseguir datos de calidad para el estudio a bajo coste?, ¿Qué es indispensable para la realización posterior de conclusiones?
  - ▶ Proporcionar un entorno único para la realización de estudios clínicos, el cual permita integrar ML y DP.

# Introducción: Objetivo

- ▶ ¿Cómo podemos conseguir datos de calidad para el estudio a bajo coste?, ¿Qué es indispensable para la realización posterior de conclusiones?
  - ▶ Proporcionar un entorno único para la realización de estudios clínicos, el cual permita integrar ML y DP.
- ▶ ¿Qué pasos debemos llevar a cabo para lograrlo?, ¿Qué herramientas son necesarias?

# Introducción: Objetivo

- ▶ ¿Cómo podemos conseguir datos de calidad para el estudio a bajo coste?, ¿Qué es indispensable para la realización posterior de conclusiones?
  - ▶ Proporcionar un entorno único para la realización de estudios clínicos, el cual permita integrar ML y DP.
- ▶ ¿Qué pasos debemos llevar a cabo para lograrlo?, ¿Qué herramientas son necesarias?
  - ▶ Estudio de los datos proporcionados.
  - ▶ Creación de un servidor para albergar los datos de los pacientes.
  - ▶ Módulo de integración de aplicaciones basadas en Maching Learning o Deep Learning con el fin de predecir el grado de enfermedad en un paciente.
  - ▶ Desarrollo de la interfaz de usuario a través de tecnologías web.
  - ▶ Implementación de los módulos estadísticos de monitorización del estudio.



# Contents

1. Introducción

**2. Herramientas**

3. Resultados

4. Conclusiones

5. Líneas Futuras

# Herramientas

- ▶ Python: Python es un lenguaje de programación a alto nivel desarrollado por Guido Van Rossum, en 1989, como continuación del proyecto programación ABC. Para el desarrollo de la aplicación ha sido básico conocer el lenguaje y sus funciones, ya que toda la aplicación está escrita en él. Además ha sido el núcleo de la unión entre las diferentes herramientas, gracias a su amplia comunidad de Software Libre.

# Herramientas

- ▶ Python: Python es un lenguaje de programación a alto nivel desarrollado por Guido Van Rossum, en 1989, como continuación del proyecto programación ABC. Para el desarrollo de la aplicación ha sido básico conocer el lenguaje y sus funciones, ya que toda la aplicación está escrita en él. Además ha sido el núcleo de la unión entre las diferentes herramientas, gracias a su amplia comunidad de Software Libre.
- ▶ Django: Es un servidor basado en Python desarrollado en el 2005 por la compañía World Company de Lawrence. Nos ha proporcionado toda el software necesario para la interacción entre máquina-base de datos y máquina-usuario.

# Herramientas

- ▶ Python: Python es un lenguaje de programación a alto nivel desarrollado por Guido Van Rossum, en 1989, como continuación del proyecto programación ABC. Para el desarrollo de la aplicación ha sido básico conocer el lenguaje y sus funciones, ya que toda la aplicación está escrita en él. Además ha sido el núcleo de la unión entre las diferentes herramientas, gracias a su amplia comunidad de Software Libre.
- ▶ Django: Es un servidor basado en Python desarrollado en el 2005 por la compañía World Company de Lawrence. Nos ha proporcionado toda el software necesario para la interacción entre máquina-base de datos y máquina-usuario.
- ▶ MongoDB: MongoDB es una base de datos no relacional, la cual fue creada en 2007 por la compañía de software 10gen, con el objetivo de competir con Google Apps Engine. En el desarrollo de nuestra aplicación es el pilar de almacenamiento de los datos, y en torno a ella se ha desarrollado la aplicación.

# Herramientas

- ▶ HTML5 y JavaScript: HTML5 y JavaScript han sido la base del desarrollo para la interacción entre usuario y servidor.

# Herramientas

- ▶ HTML5 y JavaScript: HTML5 y JavaScript han sido la base del desarrollo para la interacción entre usuario y servidor.
- ▶ CSS3 y Bootstrap: CSS3 y Bootstrap son la base del desarrollo de la interfaz del usuario para sacarle el máximo partido al servidor.



django



# Contents

1. Introducción

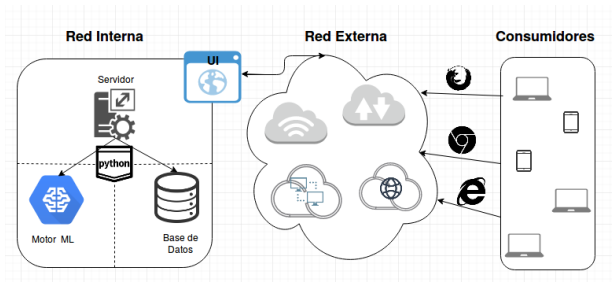
2. Herramientas

**3. Resultados**

4. Conclusiones

5. Líneas Futuras

# Resultados





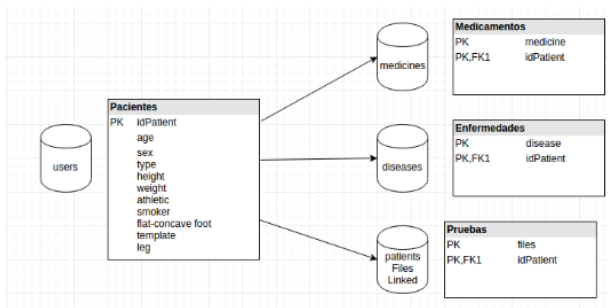






# Resultados: Base de datos desarrollada

A través del estudio de los datos llegamos a la conclusión de que lo óptimo era usar una base de datos no relacional: MongoDB.



En el siguiente diagrama ER, vemos los esquemas diseñados:

- ▶ **Pacientes:** Representa los sujetos de estudio.
- ▶ **Medicamentos:** Representa las medicinas que el paciente ingiere.
- ▶ **Enfermedades:** Representa las patologías del paciente.
- ▶ **Pruebas:** Representa los datos añadidos sobre el estudio en cuestión del paciente.

# Resultados: Servidor desarrollado

En cuanto al servidor, tenemos el que proporciona Django basado en el lenguaje de programación Python, dentro del cual se tiene en cuenta los siguientes parámetros configurables:

- ▶ Ajustes del servidor: settings.py
- ▶ Interacción Servidor Usuario: A través de las URLs del urls.py

Además para la interacción con la base de datos se disponen de una serie de APIs:

# Resultados: Servidor desarrollado

En cuanto al servidor, tenemos el que proporciona Django basado en el lenguaje de programación Python, dentro del cual se tiene en cuenta los siguientes parámetros configurables:

- ▶ Ajustes del servidor: settings.py
- ▶ Interacción Servidor Usuario: A través de las URLs del urls.py

Además para la interacción con la base de datos se disponen de una serie de APIs:

- ▶ API pacientes
- ▶ API medicamentos
- ▶ API enfermedades
- ▶ API pruebas

Y se dispone de una quinta API a parte para la integración de los gráficos:

# Resultados: Servidor desarrollado

En cuanto al servidor, tenemos el que proporciona Django basado en el lenguaje de programación Python, dentro del cual se tiene en cuenta los siguientes parámetros configurables:

- ▶ Ajustes del servidor: settings.py
- ▶ Interacción Servidor Usuario: A través de las URLs del urls.py

Además para la interacción con la base de datos se disponen de una serie de APIs:

- ▶ API pacientes
- ▶ API medicamentos
- ▶ API enfermedades
- ▶ API pruebas

Y se dispone de una quinta API a parte para la integración de los gráficos:

- ▶ API gráficos

# Resultados: Interfaz de usuario desarrollada

La interfaz desarrollada para el uso del servidor está basada en el diseño responsivo que integra bootstrap en HTML5.

Se quiere que la aplicación tenga usabilidad desde cualquier tipo de dispositivo, y que además sea fácil de usar. Para ello se han seguido unos patrones básicos de diseño:

- ▶ Simplicidad: Cuanto menos elementos que distraigan al usuario mejor.
- ▶ Homogeneidad: Un aspecto similar a lo largo de toda la aplicación.
- ▶ Coherencia u orden: Agrupación de los elementos por categorías.



# Contents

1. Introducción

2. Herramientas

3. Resultados

**4. Conclusiones**

5. Líneas Futuras

# Conclusiones

- ▶ Conocimiento biomédico adquirido: Durante el desarrollo de la aplicación he aprendido a observar, recabar, analizar, filtrar y trabajar, información adquirida de pacientes.

# Conclusiones

- ▶ Conocimiento biomédico adquirido: Durante el desarrollo de la aplicación he aprendido a observar, recabar, analizar, filtrar y trabajar, información adquirida de pacientes.
- ▶ Conocimiento médico adquirido: El conocimiento médico adquirido está relacionado con los sistemas endocrino, vascular y respiratorio, ya que durante todo el proceso de observación de los datos, estos estaban relacionados con personas que padecían enfermedades en los diferentes sistemas.

# Conclusiones

- ▶ Conocimiento biomédico adquirido: Durante el desarrollo de la aplicación he aprendido a observar, recabar, analizar, filtrar y trabajar, información adquirida de pacientes.
- ▶ Conocimiento médico adquirido: El conocimiento médico adquirido está relacionado con los sistemas endocrino, vascular y respiratorio, ya que durante todo el proceso de observación de los datos, estos estaban relacionados con personas que padecían enfermedades en los diferentes sistemas.
- ▶ Conocimiento en bases de datos no relacionales adquirido: Durante el desarrollo de la aplicación y para solventar la previsión de manejo de grandes volúmenes de datos, se aprendió MongoDB e integrarlo con Python para su uso en el servidor, además de como diseñar el modelo de datos en base de datos no relacionales.

# Conclusiones

- ▶ Conocimiento adquirido sobre ML: Sobre como interactúan las librerías más usadas: pandas, sklearn y numpy, principalmente. Saber que datos esperan y como los trabajan para el diseño de la interacción.

# Conclusiones

- ▶ Conocimiento adquirido sobre ML: Sobre como interactúan las librerías más usadas: pandas, sklearn y numpy, principalmente. Saber que datos esperan y como los trabajan para el diseño de la interacción.
- ▶ Conocimiento sobre diseño HTML: Hace referencia a la integración y diseño de los gráficos, y proporcionar una interfaz útil y completa para la monitorización y gestión de usuario.

# Contents

1. Introducción

2. Herramientas

3. Resultados

4. Conclusiones

**5. Líneas Futuras**

# Líneas Futuras

- ▶ Pruebas masivas: Antes de poder usarse como una herramienta global y abierta, se necesita de un periodo de pruebas en el cual se estrese el servidor para comprobar el correcto funcionamiento de la aplicación.



## Líneas Futuras

- ▶ Pruebas masivas: Antes de poder usarse como una herramienta global y abierta, se necesita de un periodo de pruebas en el cual se estrese el servidor para comprobar el correcto funcionamiento de la aplicación.
- ▶ Desarrollo de un estudio de ML sobre la EAP: Se pretende realizar un estudio de ML sobre la EAP, para el Trabajo de Fin de Máster, aprovechando los conocimientos adquiridos en este área durante el máster.



# MODELADO DE DATOS Y APLICACIÓN WEB PARA EL ESTUDIO DE LA ENFERMEDAD ARTERIAL PERIFÉRICA

Jorge Simón Gil



15 de diciembre de 2017