**Trabajo Final de Master**

Edgar Andrés Marín Vega

Universidad Internacional de Valencia

Desarrollo de Aplicaciones y Servicios Web

MASW10

**Índice**

[1. Introducción 1](#_Toc110280462)

[2. Contexto del Dominio 1](#_Toc110280463)

[3. Arquitectura Solución 2](#_Toc110280464)

[3.1 Persistencia 3](#_Toc110280465)

[3.2 Negocio 3](#_Toc110280466)

[3.3 Presentación 3](#_Toc110280467)

[4. Stack MEAN 3](#_Toc110280468)

[4.1 MongoDB 3](#_Toc110280469)

[4.2 Express 4](#_Toc110280470)

[4.3 Angular 4](#_Toc110280471)

[4.4 Node 4](#_Toc110280472)

[5. ETL 4](#_Toc110280473)

[6. Método Experimental 5](#_Toc110280474)

# 1. Introducción

El trabajo final aquí presentado surge de la idea de crear una aplicación que pueda tener alguna utilidad real en las personas, aprovechando mi interés personal en el tema de la salud, sobre el cual he estado profundizando durante cierto tiempo, claro está, sin perder de vista el objetivo principal de este trabajo el cual consiste en aplicar los conocimientos adquiridos durante el transcurso de la maestría.

# 2. Contexto del Dominio

La aplicación que aquí se presenta surge de una necesidad actual, relacionada con el área de la salud, en donde los conceptos médicos a nivel mundial suelen ser muy subjetivos de cada profesional, según su formación, experiencia, país, institución para la cual trabaja y hasta sus creencias.

La propuesta consiste en la de presentar una herramienta que a partir de datos reales, se pueda obtener información valiosa y subjetiva, la cual pueda servir para tener un diagnóstico más personalizado y preciso, teniendo en cuenta una mayor cantidad de criterios, en comparación con los criterios generales aplicados por la medicina general.

La herramienta que se presenta tiene 2 componentes principales, destinados a mitigar esta situación que son, un componente de evaluación de riesgo de muerte por una causa específica y un componente que permita mostrar gráficas creadas a partir de datos reales y que se obtienen de fuentes de datos públicas.

Se consideran en principio las 3 principales causas de muerte, donde cada una de ellas cuenta con diferentes métodos de evaluación de riesgo, para lo cual se creó la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| Causa de Muerte | Método | Descripción |
| **Riesgo cardiovascular** | Framingham risk score | Se obtiene a partir de un análisis de unos factores de riesgo prestablecidos |
| **Insuficiencia cardiaca** | Ejection fraction | Requiere análisis por medio de electrocardiograma |
| **Cancer** | Combinación de factores de riesgo e imágenes | Se requiere análisis invasivo para determinar con precisión. |

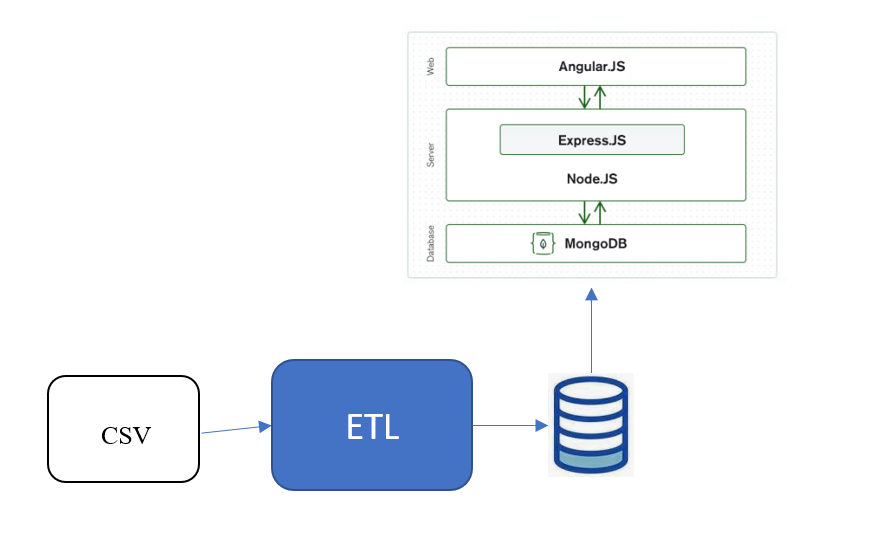
.

# 3. Arquitectura Solución

La arquitectura de la aplicación se dividirá en 3 capas que son, capa de presentación, servicios y persistencia. Se utilizará el stack MEAN, tal como se definió en la propuesta del presente proyecto. Dicho stack se compone de las siguientes tecnologías:

* **M**ongoDB (Persistencia)
* **E**xpress (Servicios)
* **A**ngular (Presentación)
* **N**odeJS (Servicios)

Mas adelante se dará una descripción en detalle de cada uno de estos componentes, los cuales se incluyen en el siguiente diagrama el cual identifica dichos componentes dentro de nuestra aplicación.



# 3.1 Persistencia

Esta capa se es la encargada de almacenar los datos de nuestra aplicación. Estos datos provienen de un proceso ETL que se encargará de extraer los datos desde fuentes de datos abiertos, los cuales podrán ser archivos CSV o servicios REST.

# 3.2 Negocio

En esta capa se crearán todos los servicios que proveerán los datos a la capa de presentación. Estos servicios contendrán la lógica de negocio de nuestra aplicación, cargar los datos desde la capa de persistencia y todas las validaciones que sean requeridas.

# 3.3 Presentación

La capa de presentación contiene las interfaces de usuario con las que el usuario final va a interactuar y que le permitirán visualizar la información deseada. Toda la lógica relacionada con la interfaz gráfica estará definida en esta capa.

# 4. Stack MEAN

# 4.1 MongoDB

Para nuestra instancia de MongoDB se utilizará el servicio en la nube MongoDB cloud, en modalidad Serverless, el cual es gratuito siempre y cuando no se exceda la cuota establecida. <https://www.mongodb.com/docs/atlas/billing/serverless-instance-costs/>

La cadena de conexión a la instancia “hearthdata” es la siguiente:



# 4.2 Express

# 4.3 Angular

# 4.4 Node

# 5. Machine Learning

# 5.1 Fuente de Datos

Para el componente de aprendizaje de máquina se utilizarán datos públicos obtenidos del National Center for Health Statistics desde el sitio <https://www.cdc.gov/nchs/ahcd/about_ahcd.htm>, de los cuales se utilizará el dataset NHAMCS (National Hospital Ambulatory Medical Care Survey) dado que es uno de los mas completos en cuanto a documentación y volumen de datos, que se encuentran en internet.

El algoritmo será entrenado utilizando datos a partir del año 2013. Se adjunta la definición original de la fuente de datos en el siguiente “Abstract”.

*This material provides documentation for users of the Emergency Department public use micro-data file of the 2013 National Hospital Ambulatory Medical Care Survey (NHAMCS). NHAMCS is a national probability sample survey of visits to hospital outpatient and emergency departments, and hospital-based ambulatory surgery centers, conducted by the National Center for Health Statistics, Centers for Disease Control and Prevention. The survey is a component of the National Health Care Surveys, which measure health care utilization across a variety of health care providers.*

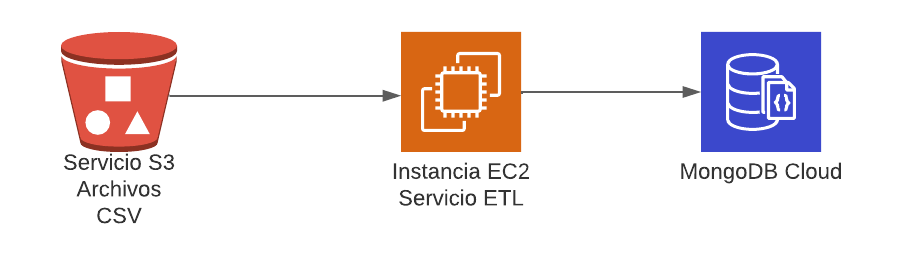
Este dataset contiene los registros que se obtienen durante el triage de un paciente a su llegada a los centros de salud. Cada registro se compone de una serie de preguntas que se realizan a todos los pacientes para lo cual cuentan con unas encuestas cuyo formato ya ha sido preestablecido. Esta serie de preguntas y sus correspondientes respuestas se pueden caracterizar en dos grandes grupos que son:

* Datos Subjetivos: Información que puede estar sujeta al criterio del personal médico
* Datos Objetivos: Corresponde a resultados de muestras de laboratorio y pruebas diagnósticas.

# 5.2 ETL

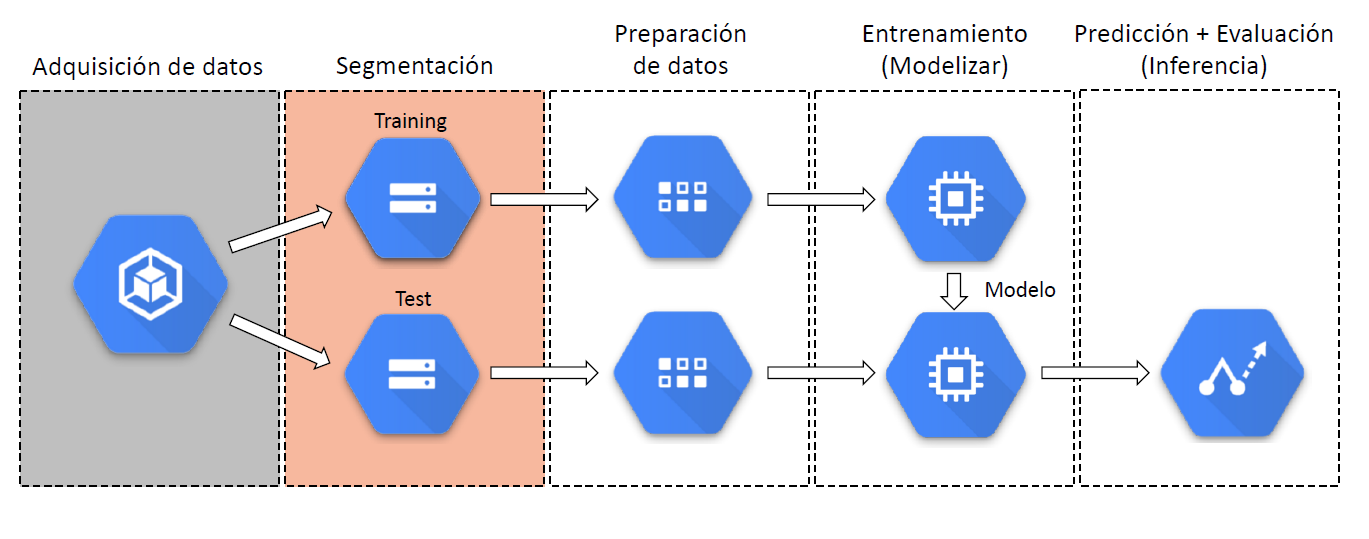
Para el componente ETL se desarrollará una aplicación en Java utilizando las librerías de SpringBoot, la cual permite crear una aplicación escalable de forma rápida y es en estas tecnologías con las cuales tengo mayor experiencia. Se utilizará la librería spring-boot-starter-data-mongodb, la cual contiene los drivers necesarios para establecer la conexión con la base de datos MongoDB y permite abstraer las operaciones de tipo CRUD que permitirán insertar los datos en la base de datos. <https://www.baeldung.com/spring-data-mongodb-reactive>.

En primera instancia se implementará el flujo de la ETL, tomando los datos desde archivos de tipo CSV los cuales se cargarán a un Bucket de S3. Para acceder a los objetos almacenados en el bucket se utilizará el SDK de Java disponible en las herramientas para desarrolladores de AWS.



# 6. Método Experimental

Definiendo un pipeline



* Algoritmos predictivos
  + Regresión logística
  + Random forest
  + Neural Network
  + Gradient Boosting