

Máster Universitario en Ingeniería Informática

Planificación y Gestión de Proyectos Informáticos

Gestión de la configuración y seguimiento de proyectos



UNIVERSIDAD DE GRANADA

E.T.S. de Ingenierías Informática y de Telecomunicación

21 de diciembre de 2020

Pablo Alfaro Goicoechea
Carlos Morales Aguilera
Carlos Santiago Sánchez Muñoz

1. Identificación de los CSCI del proyecto

Los CSCI del proyecto “DESARROLLO DE UN SISTEMA EN TIEMPO REAL DE MONITORIZACIÓN DE PACIENTES PARA SABER SU ESTADO ANÍMICO” son:

- El **código**. Nuestro proyecto conlleva implementar varios modelos de aprendizaje automático que aprendan de datos de pacientes del hospital y obtener relaciones y comportamientos.
- El **documento de requisitos** de cada uno de estos modelos.
- Los **informes de los especialistas** médicos evaluando a los pacientes y dando conclusiones para el proyecto.
- La **memoria del proyecto** que detalla las características y la trazabilidad llevada en el mismo.
- Los **datos extraídos en el hospital** a numerosos pacientes y etiquetados por los especialistas.
- El **informe del presupuesto**.
- La **planificación temporal** del proyecto (actualizada diariamente).
- **Diagramas de diseño de la arquitectura** de la aplicación en tiempo real que se va a desarrollar para monitorizar pacientes.
- Los **tests** del código (de los modelos) que garantizan la calidad del proyecto y su desarrollo.
- Los **sistemas de integración continua**.

2. Generación de la estructura de directorios del proyecto

La estructura de directorios es:

- **src**: fichero con los fuentes implementados. Cada modelo estará en un fichero .py (escrito en Python).
- **data**: carpeta con los datos extraídos de pacientes del hospital
- **docs**: diferentes documentos del proyecto.
- **test**: test de los modelos.
- **.travis.yml**: Es el fichero de configuración de nuestro sistema de integración continua, el cual garantiza un desarrollo de calidad mediante la gestión automática de test e integridad del proyecto.

```

src
├── SVC.py
├── LR.py
└── RN.py

data
└── datos.csv

docs
├── planificacion.pdf
├── requisitos.pdf
├── presupuesto.pdf
├── memoria.pdf
├── informes
│   ├── informe_1_EugenioRodriguez.pdf
│   ├── informe_2_CarlosMartinez.pdf
│   └── informe_3_BlasDiaz.pdf
├── diagramas
│   ├── diagrama_secuencia.pdf
│   ├── diagrama_estados.pdf
│   └── arquitectura_app.pdf
└── test
    ├── test_SVC.py
    ├── test_LR.py
    └── test_RN.py

.travis.yml

```

3. Tipo de sistema de control de versiones a usar

Usaremos **Git** como sistema de control de versiones. Además usaremos **GitHub** como sistema de repositorios online dado que el equipo tiene experiencia usándolo y es gratuito.

Existe la posibilidad de usar repositorios colaborativos, realizar *forks* de los repositorios y además se pueden crear repositorios privados. Asimismo, resulta cómodo el uso de *issues* en **GitHub** para diferentes tareas y unos compañeros pueden abrir *issues* a otros colaboradores para que queden reflejadas todas las modificaciones que se realicen el proyecto y notas que se deban considerar en el futuro.

4. Definición del GIT workflow

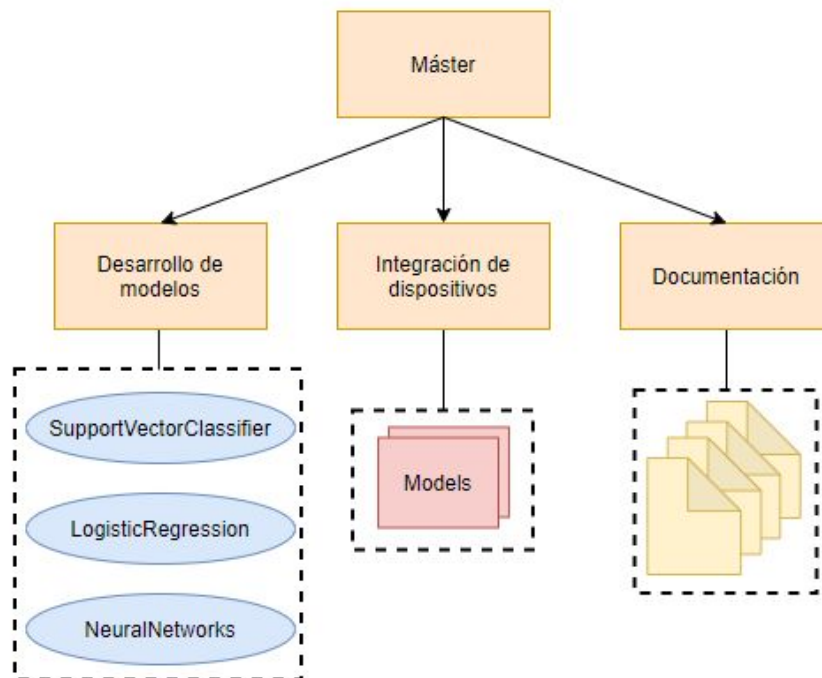
El proyecto va a tener diferentes ramas de trabajo a parte de la rama *Master* o principal.

- **Master:** Contiene todas las versiones que son desplegadas del proyecto.
- **Desarrollo de Modelos:** cada modelo de aprendizaje implementado tendrá una rama de trabajo y se mergeará a la principal cuando haya versiones estables y ofreciendo resultados sobre los datos a tratar. Por ejemplo si se van a implementar tres modelos *SupportVectorClassifier*, *LogisticRegression* y uno basado en Redes Neuronales entonces habrá tres ramas que se llamen SVC, LR y RN. Si en el futuro se planea implementar otro modelo de clasificación tendrá su rama correspondiente.
- **Integración de dispositivos:** todo lo referido al desarrollo de la aplicación en tiempo real de monitoreo mediante *wearables*.

- **Documentación:** cualquier avance en informes, presupuesto, costes, planificación, memoria, datos de los pacientes estará en dicha rama.

Nota: La integración continua se puede configurar de manera que esté presente en las ramas que se desee.

Diagrama de organización:



5. Establecimiento de la política de reuniones

Tal y como se expuso en la planificación del proyecto se realizarán reuniones bisemanales con los clientes en donde se irán mostrando los diferentes avances y/o entregas del proyecto. El cliente nos indicará si está conforme o si hay que hacer alguna modificación.

La política de reuniones del equipo de trabajo sobre el proyecto será diaria de lunes a viernes. Cada uno de los tres miembros del equipo tendrá entre 5 y 15 minutos como máximo para comentar sus avances y lo que tiene que realizar ese día. De este modo todo el equipo está conectado trabajando en una línea en común. Dichas reuniones serán a primera hora del día, a las 9:00h, ya sea *online* o presencial dependiendo de la semana y situación sanitaria.

6. Lista de aspectos que se analizarán una vez desarrollado el software

Los aspectos que se analizarán finalizado el desarrollo del software son los siguientes:

- La calidad de los modelos de inteligencia artificial desarrollados.
- Sistema de integración continua funcionando y comprobando los tests para las distintas versiones.
- Resultados de los modelos. Para que este proyecto tenga éxito es necesario un buen porcentaje de acierto en la predicción del estado de los pacientes.
- Que la aplicación en tiempo real sea capaz de monitorizar al paciente y predecir al instante algunos aspectos sobre su estado anímico o la actividad que está realizando.
- Documentación de todo el desarrollo con una memoria contando los diferentes avances y logros del proyecto.

Estos *items* son una *checklist* que se comprobará a lo largo del proyecto y muy especialmente al final para comprobar que se cumple con lo que se planteó.