ADR Gestión de la Configuración

# Contexto

## Se requiere de una herramienta donde almacenar los archivos relacionados con el proyecto que se vayan desarrollando (código fuente, documentación, etc.), así como una organización concreta para su orden y claridad.

## Procesos

* Creación de un repositorio disponible para todos los miembros del equipo.
  + Acceso a la nube por medio de navegador, línea de comandos o interfaz gráfico.
* Organización del repositorio.
  + Utilización de árbol de directorios para ubicar rápidamente los archivos.
    - Directorio "doc": almacenar toda la documentación referente al proyecto, por ejemplo, los propios ADRs.
    - Directorio "src": almacenar todo el código desarrollado.
    - No se descarta la creación de más directorios conforme las necesidades del equipo así lo requieran.
  + Utilización de ramas (*branches*) para disponer el código:
    - Rama *master*: rama principal y única.
* Política de nombres en los archivos:
  + ADRs: deben comenzar con ADR y tras un '\_' especificar con sustantivo (o varios en caso de ser necesario) el objetivo principal del mismo o tema principal tratado.
  + Pudieran añadirse más normas si así se requiriese (siempre dejando constancia).
* Acciones pertinentes a cambios en archivos:
  + Siempre que se realice un cambio, incluir un comentario breve con la descripción o motivo de lo realizado.
  + Realizar peticiones de cambios (*pull* *requests*), no actualizar sin control.
* Realización de pruebas sobre el código desarrollado:
  + Lanzar desde el acceso al repositorio (rama *master*) los tests pertinentes (tanto de integración como unitarios) para la comprobación del código desarrollado cada vez que se genere un cambio significativo.

# Opciones consideradas

1. GitHub:

* Pros:
  + Permite almacenar todos los tipos de archivos y de la forma establecida.
  + El equipo está familiarizado con su uso.
  + Permite realizar *pull requests*.
  + Se puede integrar con Trello (herramienta de gestión del equipo).
  + Disponemos de "Paquete Education":
    - GitHub Pro gratuito: repositorios ilimitados, colaboradores ilimitados, etc.
  + Posibilidad de integración con Jenkins (herramienta para integración continua).
  + El equipo tiene cuentas registradas.

1. Bitbucket:

* Pros:
  + Gratuito para equipos de hasta 5 usuarios: repositorios privados ilimitados.
  + Se puede integrar con Trello (herramienta de gestión del equipo).
  + Permite realizar *pull requests*.
  + Posibilidad de integración con Jenkins (herramienta para integración continua).
* Cons:
  + El equipo no está familiarizado con su uso.
  + No se tienen cuentas ya registradas.

# Decisión

Se utiliza la herramienta **GitHub**.

No se han planteado más alternativas debido a que hay múltiples herramientas disponibles con características muy similares (Bitbucket es un ejemplo).

Se ha decidido en favor de GitHub no sólo por lo que ofrece (que, como se dice, es bastante común al resto) sino debido a que es completamente gratuito por el "Paquete Education" del que ya disfruta el equipo. Además, ya se tienen cuentas abiertas y utilizadas en la plataforma, por lo que su uso, al menos básico, es conocido.

# Estado

(Tachar las que no apliquen)

1. Aceptado
2. ~~Rechazado~~
3. ~~Reemplazado~~

# Implicaciones

Todos los miembros del equipo deben tener una cuenta registrada en la herramienta (ya se cuenta con ello) y acceso (incluyendo permisos de edición) al repositorio utilizado para el desarrollo del proyecto.

El uso de la herramienta seleccionada debe ser ágil y preciso ya que será utilizada a diario para la actualización del trabajo.

Además, se deben tener en cuenta las normas establecidas en este ADR en cuanto a la disposición del repositorio y sus componentes.

### Fecha

27 de marzo de 2020