

Capítulo 2

Objetivos

El objetivo de este proyecto es desarrollar un módulo de analíticas para la monitorización de uso de la plataforma educativa *online* Kibotics. Se busca que esta herramienta ofrezca a los administradores de la web información sobre aquellos eventos que se consideren importantes mientras los usuarios interactúan con ella.

Para desarrollar la herramienta primero ha sido necesario estudiar, implementar y probar distintas tecnologías de recogida, almacenamiento y generación de visualizaciones. Es en estos tres puntos en los que divide el desarrollo de la herramienta.

Para asegurar que se daban los pasos correctos para la consecución de estos subobjetivos se ha establecido un plan de reuniones semanales con los tutores del proyecto. En estas se presentaban los progresos de la semana. Además, se han utilizado para resolver dudas y problemas que han ido surgiendo.

Durante la primera etapa del proyecto, se estudiaron tanto las tecnologías como la arquitectura que daba forma a Kibotics. Para mantener un control del código desarrollado, ha sido necesario el uso de un repositorio de datos, para este trabajo se han utilizado dos proyectos GitHub.

El primero ¹, un proyecto personal en el que se han subido periódicamente diferentes recur-

¹<https://github.com/RoboticsLabURJC/2019-tfg-angel-perea>

so como por ejemplo: *scripts*, ficheros de log con los que se trabajó en los momentos iniciales del proyecto, así como pequeñas pruebas sobre servidores Django. Al haber un único desarrollador el uso ha sido con *commits*.

Una vez adquirido el contexto y habiendo entendido la forma de trabajar a través de GitHub, se comenzó a trabajar en el propio código de la plataforma. Este código está almacenado en el repositorio privado de Kibotics ². Tras las reuniones semanales, se creaban *issues* que explicaban problemas o mejoras que tenía el código de la plataforma. Estos *issues* son sobre los que se trabajaba durante la semana.

Para permitir que varios desarrolladores pudiesen añadir funcionalidad o solucionar errores de forma paralela, se creaba una nueva rama actualizada con los últimos cambios de la rama principal. Es sobre esta rama sobre la que se desarrollaba la solución, una vez finalizados los cambios se explicaban en un comentario o *commit* y se subían al repositorio. Para integrarlos con la rama principal se abrían peticiones *pull request* en las que se solicita a los administradores del proyecto una verificación de los cambios subidos a la rama. Una vez aceptado el *pull request*, el proceso de integración termina y se resuelve el *issue*.

Se ha trabajado en el repositorio siempre junto a la filosofía *release often, release soon* y gracias a este sistema de ramas y pequeños parches se consigue que el código a desarrollar no sea tan extenso, permitiendo a los administradores validar e integrar los cambios más rápido.

²<https://github.com/jderobot-hub/kibotics-webserver>