

PROYECTO ASSET-O-TRON

Miembros del equipo

Nerea Díaz, Lucas Gómez, Marco Antonio López de la Nieta, Diego Sánchez y Ronald Romero

Memoria Técnica

En un primer lugar se establecieron las condiciones de trabajo del equipo en metodología agile: dos sprint con foco en funcionalidad, dos reuniones diarias a principio del día para planificación y fin del día para retrospectiva, feedback y análisis de obstáculos, micro-reuniones optativas cada 30-40 minutos de quickcheck. Con gran énfasis en cohesión y comunicación eficaz de grupo utilizando póker scrum personalizado, paneles trello y demás herramientas de productividad básica.

Posteriormente procedimos con la generación de controladores y modelos básicos para los requerimientos del proyecto. En un principio usamos el sistema de base de datos por defecto de Rails, sqlite, el que usábamos habitualmente en nuestros proyectos de clase y con el que ya estábamos familiarizados. Al avanzar en el proyecto y por requerimientos de algunas de las gemas que el mentor nos aconsejaba utilizar, sqlite dejó de ser útil y comenzamos un nuevo proyecto usando mySql, para poder usar nuestros ordenadores de desarrollo de clase, en entorno Windows. El esquema de trabajo en equipo era a través de control de versiones por repositorio distribuido (bitbucket) con ramas por persona con merge diario.

Después de muchos quebraderos de cabeza nos rendimos ante la evidencia de que si continuábamos desarrollando en Windows los tiempos de desarrollo se duplicarían y la funcionalidad de la aplicación se vería mermada en última instancia, así que decidimos comenzar, por tercera vez, de nuevo el proyecto. Esta vez en entorno Linux.

Este esquema de empezar proyectos desde cero se debe a que el trabajo distribuido habitualmente generaba bastante código no utilizable, con grandes necesidades de refactorización. Además, por otro lado, siempre centrados en el aprendizaje constante nuestros tiempos de desarrollo se habían reducido notablemente a través de ensayo-error considerando este el método más eficaz y limpio para alcanzar nuestros objetivos.

El repositorio distribuido pasó a tener una única rama principal con merge altamente controlado y desde un solo equipo y usuario administrador. El trabajo estaba orientado a objetivos funcionales y básicamente a alcanzar las fechas de entrega intermedia satisfactoriamente.

En el aspecto técnico, dadas las especificaciones requeridas por el cliente, redradix, se han utilizado las siguientes gemas:

- **Devise**: utilizada para gestionar la autenticación y autorización de usuarios.
- **Sphinx**: utilizada para realizar las búsquedas entre los recursos de la aplicación.
- **Paperclip**: utilizada para realizar la subida de archivos locales a la aplicación.

- **Acts-as-tagable-on**: utilizada para añadir etiquetas a los recursos de la aplicación con el objetivo de gestionarlos también a partir de ellas.

En cuanto a la aplicación, el usuario puede realizar las siguientes acciones, la mayoría descritas como objetivo por la empresa cliente:

- Edición de su perfil interno.
- Visualización todos los proyectos y recursos existentes.
- Creación de proyectos.
- Asignación recursos a los proyectos de los que el usuario sea miembro.
- Realización de comentarios a los recursos.
- Selección de recursos en función de etiquetas.

Finalmente vimos factible la opción de llevar una versión reducida y móvil a la presentación final, con lo que el proyecto se dividió en dos ramas, una quedaba centrada en la funcionalidad de la aplicación y la otra en el desarrollo de un maquetado más vistoso y atractivo y una funcionalidad adecuada al servidor de despliegue.

Para los ejemplos se han utilizado tanto recursos de la empresa mentora (con su permiso), como los propios, generados por las necesidades del proyecto en sí o la aplicación.