

ToDoList App

Equipo 15

Álvaro Lario
Sergio Baeza
Eugenio Benito





Índice

1. Metodología
 - a. Pair Programming
 - b. TDD
2. Conclusiones
3. Funcionalidades implementadas
4. Demo



1. Metodología

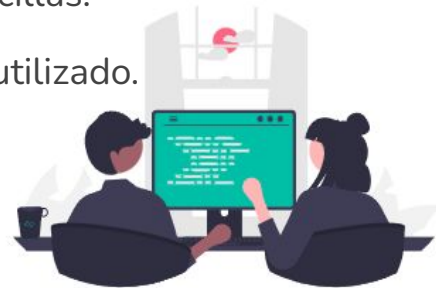


- Se ha realizado un sprint de **4 semanas** siguiendo la metodología Scrum.
- Durante el transcurso de la práctica hemos hecho uso del tablero **Kanban** para llevar un seguimiento del trabajo mejorando la **colaboración** en el proyecto por parte de sus integrantes, así como gestionar el **flujo de trabajo** para llevar un desarrollo **predecible** y **sostenible**.
- Cada semana hemos hecho una reunión para ver los progresos individuales y colectivos del equipo.
- En cuanto al desarrollo del proyecto hemos hecho uso de la herramienta **Git** y su metodología denominada **Git Flow** desarrollando cada tarea en una rama concreta y **combinando** los cambios ya programados y testeados en las ramas principales mediante uno de los flujos profesionales denominados **Pull Requests**.



1.a. Pair Programming

- Lo hemos utilizado en funcionalidades complejas como implementar calendario o administrador del equipo.
- Ha sido muy positivo a la hora de ofrecer un **punto de vista distinto**, “mente fresca”
- No es muy útil a la hora de implementar funcionalidades sencillas.
- Se ha obtenido una **mayor calidad del código** cuando se ha utilizado.
- Ha evitado el **bloqueo** a la hora de programar.





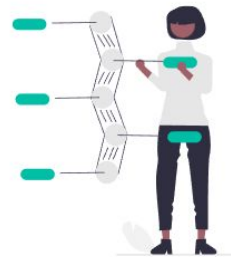
1.b. TDD

- Se ha **aplicado TDD** en todas las historias que ha sido posible
- Tests **enfocados** en **cumplir** las COS de cada historia de usuario.
- Nos ha proporcionado **tranquilidad** a la hora de añadir funcionalidades al proyecto.
- No ha sido posible en funcionalidades como el envío de correo.
- **Ralentizaba el desarrollo** debido a que la batería de tests era demasiado grande.





2. Conclusiones



- El tablero Kanban permite conocer el **estado actual** del proyecto.
- Las **estimaciones** que dimos el primer día **no han coincidido** en la mayoría. Ahora podríamos estimar mucho mejor las tareas.
- La forma de trabajar con Git Flow ha permitido el **desarrollo paralelo**.
- El revisor del PR tiene utilidad a la hora de revisar los test y la integración. Es importante detallar la historia de usuario para saber si se ha **implementado correctamente**.
- **Claridad a la hora de desarrollar** gracias a definir las historias de usuario con mockups.
- La **dependencia** entre historias de usuario nos ha marcado la **prioridad** de cada una.



3. Funcionalidades Implementadas (15 SP)

- Tareas con **fecha y vista de calendario**
- Editar **perfil** de usuario
- Administrador de Equipo
- **Comentarios** en Equipos
- Estado de progreso en las Tareas
+ **vista tablero**
- Pestaña '*Mis Equipos*'
+ Equipos administrados
- Recuperar contraseña por **correo electrónico**
- Equipos públicos o privados
+ acceso por **invitación**
- Equipos con **Proyectos**
+ Tareas de Proyecto
+ asignación de Tareas a Usuarios
- Recordatorio '*Tareas para hoy*'
- **Notificaciones** por correo electrónico



4. Demo

