



# ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

OPERA ID:125

---

## Decide Ortosia - Autenticación

---

*Autores:*

Álvaro González  
Luca Bonacini  
Antonio Millan  
Rodrigo Rojas  
Antonio Romero

*Tutores:*

David Benavides Cuevas  
*Opera Id:* 125  
*Grupo:* G1

## Índice

<b>1. Resumen</b>	<b>2</b>
<b>2. Introducción y contexto</b>	<b>3</b>
<b>3. Descripción del sistema</b>	<b>3</b>
<b>4. Planificación del proyecto</b>	<b>3</b>
<b>5. Entorno de desarrollo</b>	<b>3</b>
<b>6. Gestión del cambio, incidencias y depuración</b>	<b>3</b>
6.1. Gestión del código fuente . . . . .	3
6.2. Gestión de la construcción e integración continua . . . . .	3
6.3. Gestión de liberaciones, despliegue y entregas . . . . .	3
<b>7. Mapa de herramientas</b>	<b>3</b>
<b>8. Ejercicio de propuesta de cambio</b>	<b>3</b>
<b>9. Conclusiones y trabajo futuro</b>	<b>3</b>

## 1. Resumen

Decide es una herramienta Open Source que se desarrolla con el fin de facilitar el voto online de manera segura, anonima y eficaz. Decide es un software heredado realizado por el equipo de Wadobo <https://github.com/wadobo/decide>. En un primer momento se realizó un fork de este sistema para que podamos trabajar de forma aislada sin tener que modificar el sistema original. El sistema se divide en varios subsistemas para facilitar el escalado y el mantenimiento del software. Dichos subsistemas se comunican entre ellos a través de llamadas REST.

Nuestro grupo se encarga del subsistema de autenticación. Es el primer paso para que un usuario pueda votar, garantizar que la persona que va a realizar el voto es quien dice ser. Como se ha comentado previamente el código fuente es heredado por lo que teníamos que seguir el lenguaje, estrategia y estructura que se había definido en un principio. El equipo decidió no desarrollar el subsistema con otra tecnología ya que podría afectar al buen funcionamiento del sistema y podríamos encontrarnos con conflictos con los resto de subsistemas.

Después el equipo llegó a un acuerdo sobre las tareas que se iban a desarrollar a lo largo del proyecto entre los que se encuentran distintos métodos de autenticación, documentación de la API, interfaz de usuario, etc. Cada miembro del equipo tenía asignadas unas tareas (*issues*) y para gestionar el código se ha usado un repositorio GIT con el fin de que tuviéramos siempre la última versión del sistema (tanto por nuestra parte como con el resto de los equipos) y poder resolver conflictos de una forma más eficiente.

2. Introducción y contexto
3. Descripción del sistema
4. Planificación del proyecto
5. Entorno de desarrollo
6. Gestión del cambio, incidencias y depuración
  - 6.1. Gestión del código fuente
  - 6.2. Gestión de la construcción e integración continua
  - 6.3. Gestión de liberaciones, despliegue y entregas
7. Mapa de herramientas
8. Ejercicio de propuesta de cambio
9. Conclusiones y trabajo futuro