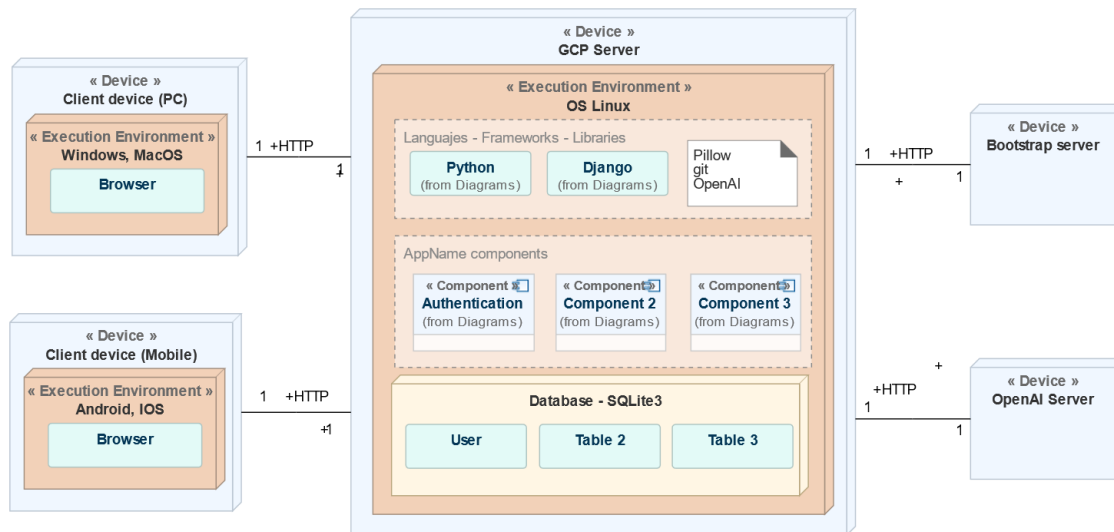


Guía para la actualización de los diagramas

Diagrama de despliegue



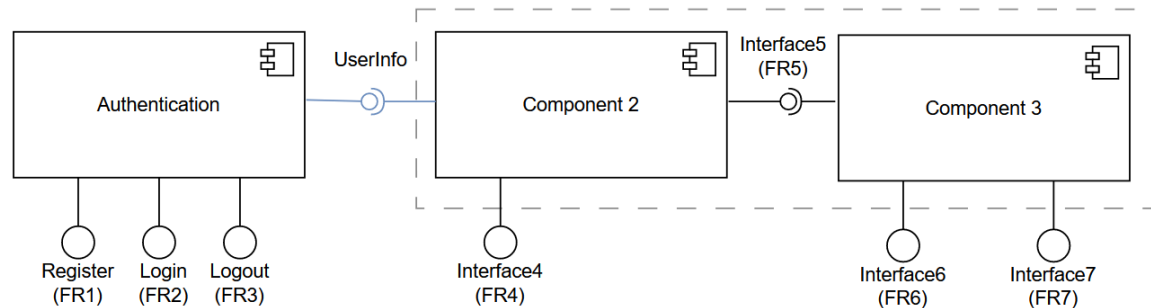
En el diagrama de despliegue se deben mostrar al menos dos dispositivos: el cliente (a la izquierda) y el servidor (en el centro), donde se desplegará la aplicación.

El dispositivo cliente es aquel desde el cual los usuarios se conectarán a la aplicación. Por esta razón, debe incluir un sistema operativo adecuado para PCs y dispositivos móviles. En este caso, se decidió crear un nodo independiente para cada tipo de dispositivo. Dentro del sistema operativo debe haber un navegador web a través del cual se accederá a la aplicación.

El dispositivo servidor, en este caso denominado 'GCP Server', es aquel en el que se despliega la aplicación. Como se utilizará Google Cloud Platform (GCP) para el despliegue, se debe especificar que el sistema operativo es Linux. Además, es necesario listar los lenguajes de programación utilizados (solo aquellos que se deban instalar), el framework de desarrollo web y las bibliotecas empleadas. En la línea intermedia, se deben listar los componentes de la aplicación, asumiendo en este caso la existencia de un componente de autenticación. Finalmente, se debe especificar la base de datos utilizada, en este caso SQLite3, y el listado de tablas que se crearán en la aplicación.

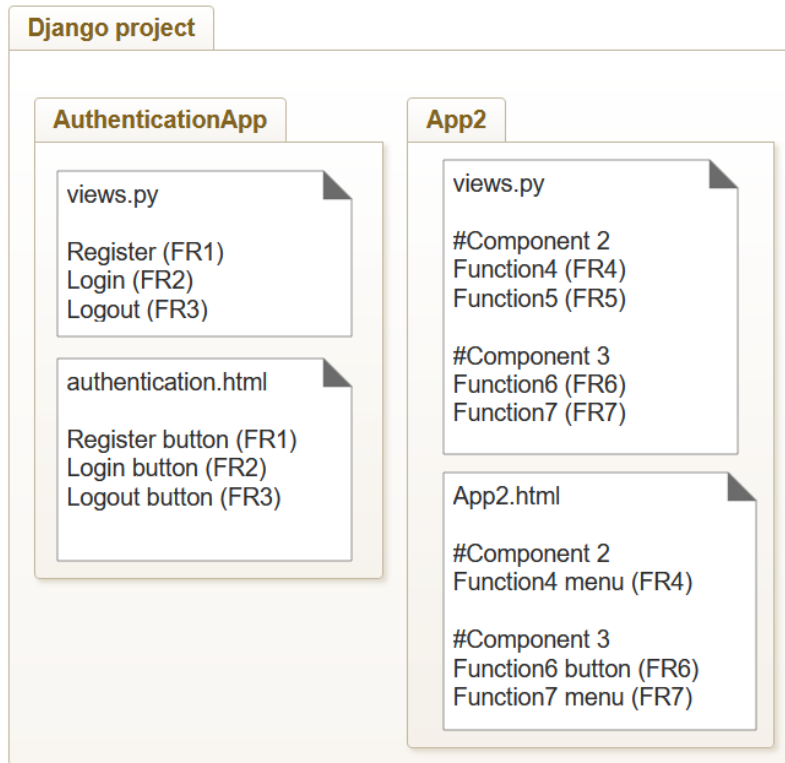
Se representarán los dispositivos OpenAI server y Bootstrap server sin detalles al interior dado que no se conocen.

Diagrama de componentes



El diagrama de componentes debe especificar claramente cómo se distribuyen los requisitos funcionales entre los distintos componentes y cuál es la relación entre ellos. En el ejemplo proporcionado, se destacan tres componentes principales: 'Authentication', 'Component 2' y 'Component 3', mostrando cómo se interrelacionan. Por ejemplo, el componente 'Authentication' aborda la implementación de los requisitos funcionales FR1 – Register, FR2 – Login y FR3 – Logout. Además, se ilustra cómo 'Authentication' transfiere la información 'UserInfo' a 'Component 2'. Adicionalmente, 'Component 3' requiere la interfaz 'Interface 5' ofrecida por 'Component 2'. Note que las interfaces representan los requisitos funcionales de la aplicación y en ocasiones representan datos necesarios para el funcionamiento de los componentes. Las interfaces que quedan sin conexión con otros componentes, por ejemplo, Register, Login o Logout, son funcionalidades que el usuario final debería percibir explícitamente en la página web (por medio de un botón, de un menú, de una opción, etc). Los componentes que están agrupados por medio del rectángulo en línea punteada están estrechamente relacionados entre ellos, por ejemplo, pueden estar la misma aplicación o pueden consultar datos de la misma tabla de la base de datos.

Implementación



Este diagrama representa la estructura del proyecto Django y su correspondencia con el diagrama de componentes. Se observa que el proyecto consta de dos aplicaciones: una aplicación está vinculada directamente al componente 'Authentication', y la otra agrupa a los componentes 'Component 2' y 'Component 3'. En el archivo `views.py` de cada aplicación debe existir una función que implemente cada requisito funcional (que se representó como interfaz en el diagrama de componentes). En el archivo `html` de cada app deben aparecer, por lo menos, las interfaces de las funcionalidades a las que el usuario puede acceder directamente (es decir, las interfaces del diagrama de componentes que no están conectadas con otros componentes). De este modo, se establece una conexión directa entre los requisitos, los componentes y la estructura del proyecto Django.