

PRUEBA TÉCNICA

El tiempo estimado de ejecución de la prueba es de **4 horas**, se podrá utilizar más tiempo si el candidato lo desea.

Dudas o preguntas por correo electrónico a:

- pablopantoja@useit.co
- andresbarbosa@useit.co
- luisrosero@useit.co

PROBLEMA:

Mini Chat Grupal

REQUERIMIENTOS:

- Se requiere que las personas se puedan registrar
- Se requiere que las personas puedan acceder a través de un Login
- Solo va existir una sala de chat en donde se conectaran todos los usuarios registrados
- Los usuarios van a poder eliminar sus mensajes

BACK-END:

Se requiere lo siguiente:

- Desarrollar el Back-end en **Django**
- Idealmente usar como Base de Datos **PostgreSQL** (Se puede usar SQLite)
- Realizar una RestFul API (**Django REST Framework**)
- **Front-end:** Vista basada en templates de Django
 - Login
 - Registro
 - ◆ Chat
 - En una columna mostrar los usuarios registrados, con la última fecha de inicio de session(síncrono)
 - Se debe hacer una API RESTFul para consultar los mensajes del chat(asíncrono)
 - Debe hacer un formulario django para hacer el envío del mensaje(el mensaje debe permitir subir unicamente imagenes)(síncrono)
 - Se debe hacer una API para eliminar los mensajes(asíncrono)
 - ◆ Idealmente utilizar Bootstrap
 - ◆ **Opcionales:**
 - Utilizar SASS o LESS

FRONT-END:

Se requiere lo siguiente:

- **Back-end:** Se podrá simular el back-end con un **Fake Server** con Firebase, Postman u otro.
- **Entregable:**
Aplicación Móvil (Ionic o React Native) o Aplicación Web (Angular o ReactJS)
- **Funcionalidades:**
 - ◆ Login
 - ◆ Registro
 - ◆ Chat
 - Lista de de usuarios registrados, con su último dato de session
 - Lista de los mensajes del chat(mostrar media)
 - Eliminar mensajes propios.
 - Crear mensajes(permitir subir imagenes)
 - ◆ **Opcionales:**
 - Utilizar SASS o LESS

OPCIONALES (PUNTOS EXTRA):

- El chat paginado, con scroll
- Los usuarios con paginación asíncrona.
- La Aplicación Web y Back-end desplegados en alguna infraestructura (AWS, GCP, Azure, etc)
- Ambiente en Contenedor (Docker), la Base de Datos debe ser persistente.
- (CI / CD) Despliegue mediante Pipeline desde el repositorio (Gitlab, Github, Bitbucket, etc)

ENTREGABLES:

- Url del Repositorio Git de la Aplicación Web.
- Url de la Aplicación Web para acceder (Opcional).
- backup sql
- Usuarios para acceder a probar.

