

PRUEBA TÉCNICA

El tiempo estimado de ejecución de la prueba es de 2 días, en caso de necesitar más tiempo o de tener alguna dificultad por favor notificar a Luisa Muriel. Dudas o preguntas por correo electrónico a:

→ andresbarbosa@useit.co

→ luisrosero@useit.co

→ luisamuriel@useit.co

Ejecución incluidas en el Readme

PROBLEMA:

El objetivo es crear un sistema de control de acceso a sedes (puntos de acceso) para personal de empresas.

El sistema debe permitir crear y controlar accesos para los empleados a las sedes de cada empresa.

El sistema consta de 2 partes:

1-Aplicación web Django para gestión.

1. Usuarios.
2. Empresas.
3. Puntos de acceso dentro de la empresa.
4. Franjas horarias de acceso por empleado.

Se consideran 3 tipos de usuarios:

1. Usuario superusuario o administrador de la aplicación (solo uno)
2. Usuario administrador de empresa.
3. Usuario empleado de empresa.

La aplicación web debe permitir:

1. El usuario administrador(superusuario) de la aplicación puede crear empresas.
2. El usuario administrador(superusuario) de la aplicación puede crear un usuario administrador para cada empresa creada e invitarlo por correo electrónico.
3. El usuario administrador de cada empresa debe poder gestionar puntos de acceso de la empresa (sedes).
4. El usuario administrador de la empresa puede invitar a un empleado de la organización a

registrarse con un vínculo dentro de un correo electrónico, el usuario se registra como perteneciente a la empresa.

Este vínculo debe llevar a un formulario de registro donde el usuario llenará sus datos básicos.

5. Cuando el usuario administrador de la empresa ingrese con su usuario y contraseña se le debe presentar una página inicial con los puntos de acceso de la empresa presentados en una grilla de 3 fichas por fila con paginación.
6. El usuario administrador de la organización puede asignar a un usuario empleado de su empresa a un punto de acceso (sede) y debe poder especificar varias franjas horarias dentro de la cual dicho empleado podrá acceder a la sede.
7. El administrador debe poder activar o desactivar accesos a cada sede.

2-REST api de consulta (POR EMPRESA).

1. Validar permisos de acceso a un punto de una empresa para sus usuarios en la hora y localización actual (sede) desde donde se está consultando.
2. Retorna true/false junto a los datos del punto de acceso (sede).
3. La validación debe considerar el estado activo o inactivo del punto.
4. Debe enviar cada intento de ingreso fallido al correo electrónico del administrador de la empresa.
5. La prueba de la API se puede hacer mediante POSTMAN.

REQUERIMIENTOS:

→ Registro e ingreso de usuarios

→ CRUD Empresa

- ◆ Usuario administrador (fk)
- ◆ Nit
- ◆ Nombre de la empresa
- ◆ Nombre comercial de la empresa
- ◆ Dirección
- ◆ Teléfono
- ◆ Correo electrónico (Debe ser un correo electrónico válido)
- ◆ Sitio web (Debe ser una dirección Web válida)
- ◆ País, Estado y Ciudad

→ CRUD Usuarios

- ◆ Apellidos
- ◆ Nombres
- ◆ Empresa (fk)
- ◆ Dirección
- ◆ Teléfono
- ◆ Correo electrónico (Debe ser un correo electrónico válido)

- ◆ País, Estado y Ciudad

→ CRUD puntos de acceso de la empresa

- ◆ Nombre
- ◆ Dirección
- ◆ Correo electrónico
- ◆ Empresa (fk)
- ◆ Geolocalización
- ◆ Estado
- ◆ Horarios de acceso (fk)

→ CRUD horarios de acceso (franjas horarias de acceso)ç

- ◆ Punto de acceso (fk)
- ◆ Hora de inicio
- ◆ Hora de finalización
- ◆ Usuario (fk)

NOTA: Estas son definiciones iniciales de modelos, es probable que debas complementarlos.

OPCIONALES (PUNTOS EXTRA):

1. Validar los teléfonos con indicativo y su localización (+57 316 878 66 28 es un teléfono de Colombia).
2. Usar un tercero (Ejemplo: Google Maps) para determinar el País, Estado y Ciudad
3. Realizar el proceso de registro de un usuario con verificación de correo electrónico, mediante el envío de un enlace con un único token.
5. Usar Docker para el despliegue.

FRONT-END:

Se requiere lo siguiente:

NOTAS:

Si solo se presenta prueba front-end, se podrá simular el back-end con un Fake Server con Firebase, Postman u otro.

→ Desarrollar el Back-end en Django (solo si presenta prueba backend)

- ◆ Idealmente usar como Base de Datos PostgreSQL (Se puede usar SQLite)
- ◆ Realizar una RestFul API (Django REST Framework)

ENTREGABLE:

Aplicación Móvil (Ionic [Angular, React o ViewJS], React Native o Flutter) o Aplicación Web (Angular, ReactJS o ViewJS)

ALCANCE:

◆ Usuarios:

◆ Ingreso con usuario

- Una vista inicial del administrador de la empresa para mostrar los puntos de acceso (sedes) de la empresa.
- Una vista al administrador de la empresa (se usa este único usuario administrador en todas las sedes) para ver detalles de punto de acceso y botón para lectura de código QR sobre la pantalla del celular del usuario empleado de la empresa. Tan pronto se lee exitosamente el código QR, se debe validar el acceso mediante la API de consulta de acceso.
- Una vista inicial para el empleado de la organización para desplegar código QR para presentar en el punto de acceso y que pueda ser leída desde el celular del administrador.
- El código QR del empleado debe incluir el identificador de usuario, y su posición actual (latitud/longitud) leída desde el celular.

◆ Idealmente utilizar Bootstrap (si aplica)

◆ Opcionales:

- Utilizar SASS o LESS (si aplica)