Prueba técnica para desarrolladores Python Backend.

Motivación

Una de las principales actividades del área de Backend dentro de guane es la construcción y consumo de API Rest para coordinar el funcionamiento de nuestras plataformas. De esta manera, los desarrolladores tienen una base sólida en HTTP, construcción de APIs siguiendo REST, uso de Docker y Celery. La prueba técnica propuesta involucra todos estos aspectos utilizando Python a la vez que FastAPI.

Arquitectura de la solución completa

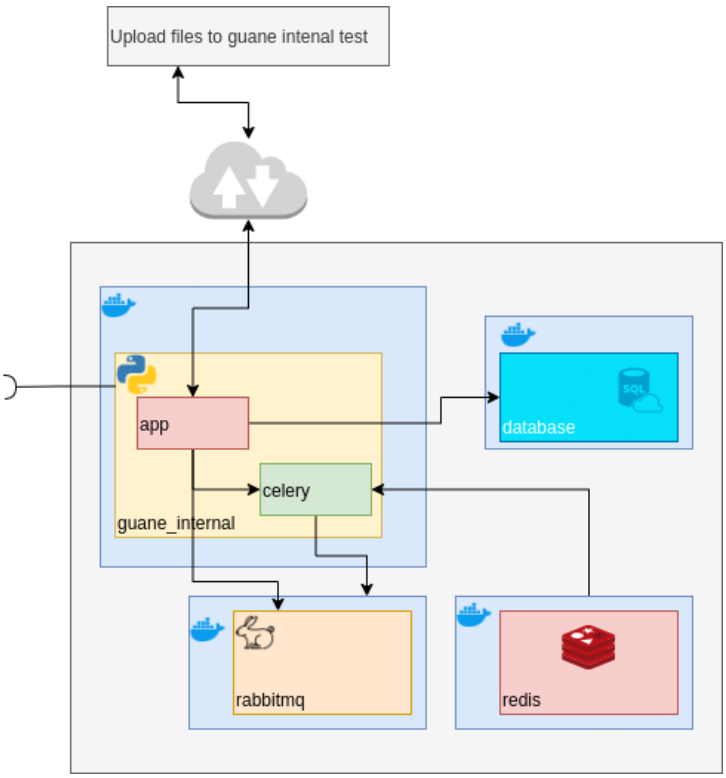


Figura 1. Arquitectura de la solución

Como puede ver la aplicación consiste en un Stack basado en las tecnologías antes mencionadas. La idea principal es desarrollar una API que siga los principios REST con un componente arquitectónico adicional que incluye realizar un llamado a una tarea asíncrona.

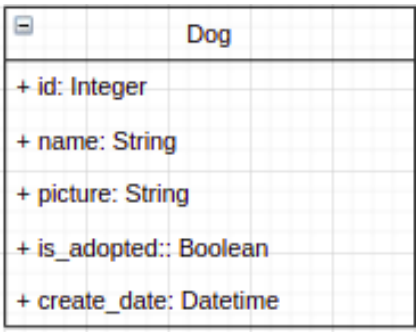
Prueba técnica Backend

Diseño de una API REST con el framework FastAPI de Python

Para esta prueba se requieren conocimientos en Python, protocolo HTTP, verbos HTTP (GET, POST, PUT (PATCH), DELETE). Haciendo uso del marco de desarrollo para Python FastAPI, se desea que el participante desarrolle las siguientes funcionalidades:

1. Obtener un listado de <entidad>
2. Obtener la información de <entidad> por su identificador unico id
3. Actualizar la información de una <entidad>
4. Eliminar la <entidad>

La <entidad> con la que vamos a trabajar tiene el siguiente modelo de datos:



La entidad que vamos a manejar en esta ocasión es Dog. Para cumplir con las necesidades enumeradas en las funcionalidades se requiere tener los siguientes Endpoints:

GET -> /api/dogs : Obtener un listado.
GET -> /api/dogs/{name} : Obtener una entrada a partir del nombre.
GET -> /api/dogs/is_adopted : Obtener todas las entradas donde la bandera is_adopted sea True.
POST -> /api/dogs/{name}: Guardar un registro según el esquema de arriba. El campo picture se debe rellenar con una consulta al API externo https://dog.ceo/api/breeds/image/random.
PUT -> /api/dogs/{name}: Actualizar un registro según el nombre.
DELETE -> /api/dogs/{name}: Borrar un registro según el nombre.

Un ejemplo de registro:

{
“id”: 1,
"name": "Lazy",
"picture": "https:\/\/images.dog.ceo\/breeds\/papillon\/n02086910_6483.jpg",
"create_date": 2020-02-20 20:58:55.164954
"is_adopted": True
}

Los datos pueden ser guardados utilizando una base de datos (la que el participante desee). Además se deben realizar los siguientes puntos.

1) Las rutas para ingresar un nuevo canino (el POST) debería estar protegida con alguna política de seguridad (recomendamos usar JWT).

2) Realizar Dockerfile y Docker-compose de la aplicación.

3) Desarrollar una tarea asíncrona con Celery cuando se llame la función POST. Las tareas en segundo plano tienen sentido para actividades que se pueden realizar de forma asíncrona. Puedes simular una latencia en el registro de los perros de unos segundos para verificar que el "Worker" o “tarea” está funcionando, o hacer uso del endpoint /api/workers disponible en link.

4) Añadir una nueva entidad llamada User con sus respectivos endpoints de CRUD básico. Esta entidad tendrá una relación de uno a muchos con la entidad Dog, es decir, un User puede tener uno o muchos Dogs (para esto solo se requiere que la entidad Dog se agregue un nuevo campo llamado id_user y guarde el id de un usuario). Los campos para la entidad User pueden ser: id, name, last_name, email.

5) Enviar un archivo con cualquier extensión al endpoint /api/files disponible en este link.

Finalmente,el código debe estar en un repositorio remoto (GitHub, GitLab, BitBucket) llamado "guane-intern-fastapi". También debes seguir el estándar del PEP8 en la codificación con lineamientos de buenas prácticas.

Notas:

- La documentación de FastAPI está diseñada como un tutorial paso a paso. Te recomendamos revisar constantemente al momento de construir el API. FastAPI utiliza Python moderno para ejecutarse, de esa manera, recomendamos usar una version superior a Python 3.8.Las bases de datos que ut
- Puedes usar la base de datos que prefieras. Nosotros utilizamos Postgresql apoyados de Tortoise.