

Diseño de las entidades

Usuarios

```
{
  "telefono": "string",
  "alias": "string",
  "contactos": [{ "alias": "string", "telefono": "string" }]
}
```

Mensajes

```
{
  "_id": "ObjectId",
  "timestamp": "datetime(format=ISO-8601)",
  "origen": "string",
  "destino": "string",
  "destino": "string",
  "texto": "string(max=400)"
}
```

Limitaciones sobre la solución propuesta

- La entrega contiene ficheros que pueden ser incompatibles con su máquina (concretamente `__pycache__` y `.venv`). Se recomienda eliminarlos antes de ejecutar el código, o bien utilizar el contenedor de Docker que se proporciona
- Las variables de entorno deberían estar ya configuradas, tanto para el `docker-compose` como para el entorno de desarrollo. En el caso del entorno de desarrollo, será necesario proporcionar una base de datos MongoDB
- El fichero `runme.sh` es un script para `zsh` utilizado para generar algunos ficheros antes de empezar el proyecto. No se recomienda ejecutar
- La carpeta `iweb` contiene una copia local de las entidades de mongo listas para ser cargadas

Instrucciones de despliegue y uso

Ejecución de desarrollo

Requisitos

- Python 3.11 o superior
- Pip3

Instrucciones

```
# Crear un entorno virtual
python -m venv .venv
source .venv/bin/activate
# Instalar los paquetes necesarios
pip install -r requirements.txt

# Configuración necesaria para arrancar el servicio
export mongo_url=<VALOR>
export mongo_collection=<VALOR>
export mongo_database=<VALOR>
# Iniciar el servidor
uvicorn --reload --port 8000 --host 127.0.0.1 src:app
```

Ejecución mediante docker

```
# Compilar el contenedor
docker build -t fastapi-app .
# Inicializar el contenedor
docker run -p 8000:8080 \
    -e mongo_url=<VALOR> \
    -e mongo_collection=<VALOR> \
    -e mongo_database=<VALOR> \
    fastapi-app
```

Ejecución mediante Docker compose

```
# Iniciar el servicio completo
# Disponible en http://localhost:8000
# Mongo en mongodb://iweb:strongpassword@localhost:27017
```

```
docker compose up -d
# Detener el servicio
docker compose down
```

Configuración

La aplicación admite las siguientes opciones de configuración mediante ficheros `.env` o variables de entorno

Variable	Descripción	Valor por defecto
<code>mongo_url</code>	URL de un servidor MongoDB	
<code>mongo_usuario</code>	Colección donde almacenar los datos sobre los usuarios	
<code>mongo_mensaje</code>	Colección donde almacenar los datos sobre los mensajes	
<code>mongo_database</code>	Base de datos donde buscar la colección	

Documentación

Se proporciona un fichero `openapi.json` con la especificación de OpenApi. Además, el propio servidor web proporciona la documentación sobre los endpoints REST bajo las rutas `/docs` (SwaggerUI) y `/redoc` (Redoc)