

Reto de Desarrollo Backend – Python

¿En que consiste el reto?

La realización de esta prueba se basará en hacer un API Rest según las características que se especifiquen.

Para ello se necesitará crear un repositorio online en el servicio que se prefiera (Gitlab, Github...) donde se subirá el código resultante de la prueba junto con los archivos que sean necesarios.

El lenguaje a utilizar como base será Python (última versión). La prueba consiste en crear un API Rest en el framework Flask utilizando los siguientes repositorios como base de datos:

URL <https://api.chucknorris.io/>



URL <https://icanhazdadjoke.com/api>



Ten en cuenta...

Se evaluará el funcionamiento de cada uno de los requerimientos, además de la claridad del código, la estructura del repositorio, pruebas unitarias, la cantidad de endpoints realizados y la documentación hecha.

Si pasas del tiempo limite se descartará la revisión del reto.

Si te sobra tiempo, tenes libertad para añadir funcionalidades extra. Lo valoraremos de manera muy positiva.

Tiempo máximo de entrega:

 1 día

¿Que buscamos?

1. Tu capacidad de análisis
2. Tu calidad de codificación
3. Tu conocimiento de Python y API REST
4. Tus conocimientos de repositorios de datos SQL y noSQL
5. Tu velocidad de desarrollo y despliegue

Tareas a realizar

1 Crear el API REST con los siguientes endpoints y su diseño en yaml para swagger:

ENDPOINT DE CHISTES

GET: Se devolverá un chiste aleatorio si no se pasa ningún path param.

- Si se envía el path param habrá que comprobar si tiene el valor "Chuck" o el valor "Dad".
- Si tiene el valor "Chuck" se conseguirá el chiste de este API <https://api.chucknorris.io>.
- Si tiene el valor "Dad" se conseguirá del API <https://icanhazdadjoke.com/api>.
- En caso de que el valor no sea ninguno de esos dos se devolverá el error correspondiente.

POST: guardará en una base de datos el chiste (texto pasado por parámetro)

UPDATE: actualiza el chiste con el nuevo texto sustituyendo al chiste indicado en el parámetro "number"

DELETE: elimina el chiste indicado en el parametro number.

ENDPOINT MATEMÁTICO

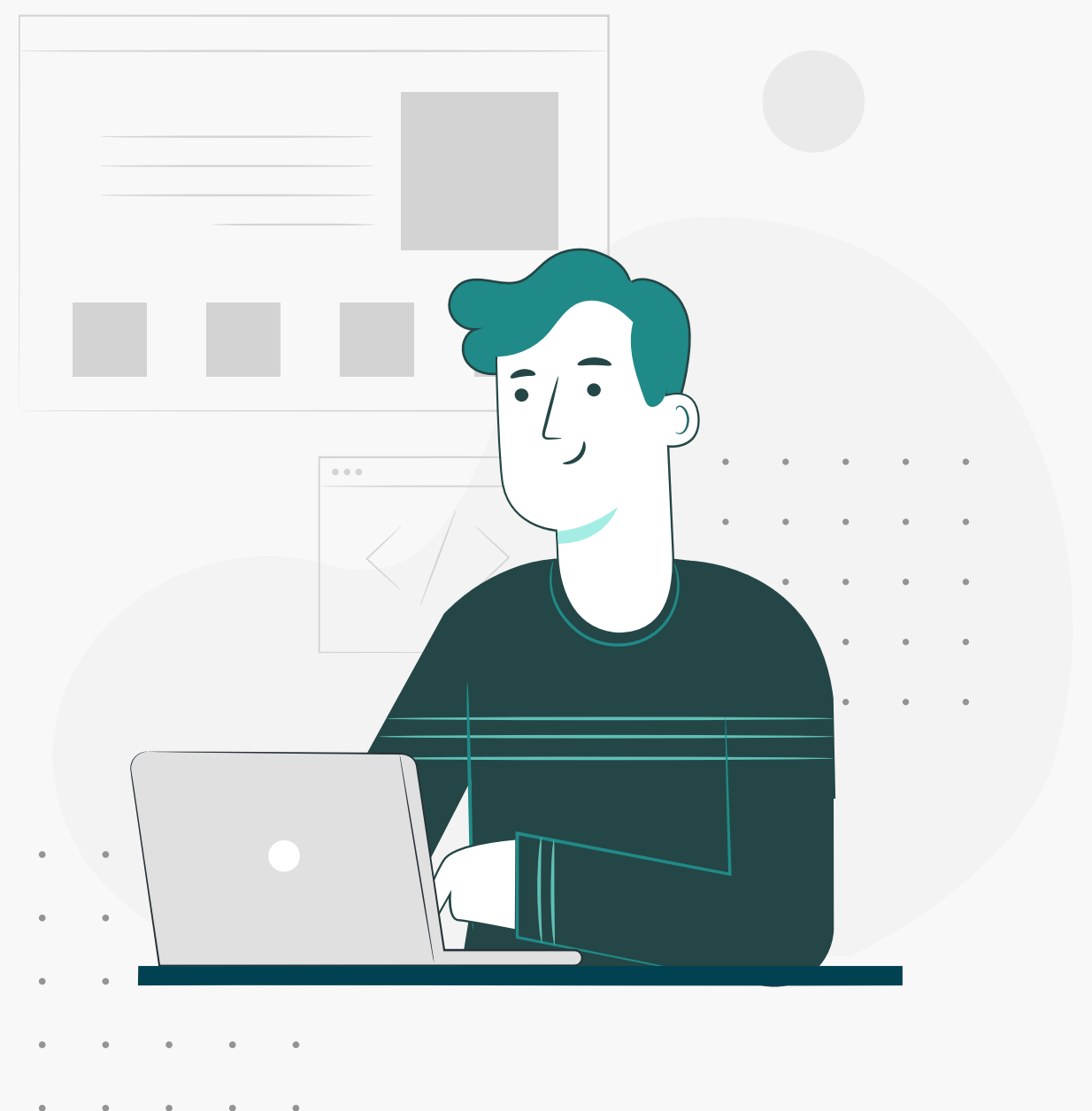
GET: Endpoint al que se le pasará un query param llamado "numbers" con una lista de números enteros. La respuesta de este endpoint debe ser el mínimo común múltiplo de ellos.

GET: Endpoint al que se le pasará un query param llamado "number" con un número entero. La respuesta será ese número + 1.

2 ¿Qué repositorio utilizarías?

PostgreSQL, MariaDB, Casandra, MongoDB, ElasticSearch, Oracle, SQL Server

1. Razona tu respuesta
2. Crea la sentencia para crear la BBDD y el modelo de datos que requerirías
3. Lo mismo que el punto anterior (si lo hiciste con una SQL) pero para un repositorio noSQL.



Entregables

Antes de que finalice el tiempo máximo de desarrollo del reto, envía la URL del proyecto público a la dirección de email: retos@squadmakers.com



Código Python

A través de un repositorio público Git



Archivo README.md

Donde se explique como has ejecutado el proyecto



Documentación

De los endpoints

¿Qué pasará después?



Ten paciencia

Nos contactaremos contigo vía correo electrónico, en el rededor de una semana



Recibe Feedback

Te compartiremos una rúbrica de evaluación con tus resultados, si aprobaste o no el reto



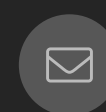
Publica tu certificación

En caso de aprobar el reto, te enviaremos un certificado digital y un badgr que podrás adjuntar en tu perfil de LinkedIn

Nota: El código que desarrolles bajo esta prueba pasará a ser propiedad de SquadMakers.



¡Buena suerte!



¿Tienes dudas? Escríbenos
retos@squadmakers.com