Code Challenge Analistas de Desarrollo

Nota: Esta prueba está contemplada para validar la capacidad de aprendizaje en un corto tiempo del candidato. Es importante mencionar que no finalizar la prueba en su totalidad no es causa de no tenerla en cuenta y evaluarla, inclusive de la posibilidad de ser seleccionado. Entre más cumpla los criterios más puntos suma y se analiza de esa forma la prueba. El tiempo para desarrollar esta prueba es de 8 días hábiles.

bvc cuenta con muchas plataformas de negociación y de tecnología para soportar y administrar el mercado de valores, a2censo es una de ellas, esta es una plataforma de financiación colaborativa, la cual permite que las Pymes colombianas obtengan y financien recursos económicos a través de campañas.

Objetivo:

Este challenge consiste en desarrollar un servicio web que exponga un endpoint para consultar la información de las campañas de a2censo, las cuales deben estar almacenadas en una base de datos. El servicio debe retornar las campañas de forma ordenada por la cantidad de recursos o recursos solicitados.

- ### Microservicio
    - Se debe realizar un microservicio usando lenguajes de programación java o python con los frameworks de preferencia como spring boot en el caso de java o django o flask en el caso de python (el framework es elección del candidato).
    - Se debe exponer como mínimo el endpoint REST API que implemente un método GET que debe permitir consultar la información de la base de datos.
    - Para probar el funcionamiento del microservicio se recomienda hacer uso de la herramienta Postman (https://www.postman.com/)
    - La respuesta del servicio lo puede hacer de la manera que sea de su preferencia. Ejemplo formato JSON.
    - Se debe entregar el servicio en un contenedor de Docker que incluya las dependencias requeridas para poder internamente en bvc correr el ejercicio y analizar el código y funcionamiento. Si tiene impedimentos con Docker puede igualmente entregar el código fuente en un repositorio GitHub público para poder realizar la valoración de prueba.

Input microservicio

Se debe recibir como parámetros el criterio de ordenamiento (amount o requestedAmount) y el tipo de ordenamiento (mayor a menor o menor a mayor)

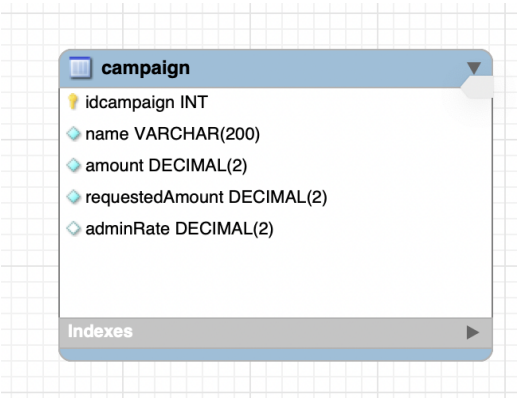
Output microservicio

El resultado esperado es un arreglo ordenado según el criterio recibido en el input

- ## Base de datos

    - Se debe usar una instancia de base de datos relacional de mysql como contenedor en un docker. https://hub.docker.com/_/mysql. Si tiene impedimentos con Docker puede tener la base de datos localmente.
    - La base de datos debe contener la siguiente tabla

      ```
      CREATE TABLE IF NOT EXISTS 'a2censo'.'campaign' (
        'idcampaign' INT NOT NULL AUTO_INCREMENT,
        'name' VARCHAR(200) NOT NULL,
        'amount' DECIMAL(2) NOT NULL,
        'requestedAmount' DECIMAL(2) NOT NULL,
        'adminRate' DECIMAL(2) NULL,
        PRIMARY KEY ('idcampaign'))
      ENGINE = InnoDB
      ```



INSERT INTO 'a2censo'.'campaign' ('name', 'amount', 'requestedAmount') VALUES ('RobinFood 2.0', 200000000, 250000000);

INSERT INTO 'a2censo'.'campaign' ('name', 'amount', 'requestedAmount') VALUES ('T4 Tea For U', 1000000000, 1200000000);

INSERT INTO ‘a2censo’.’campaign’ (‘name’, ‘amount’, ‘requestedAmount’) VALUES (‘Smoking Burgers’, 200000000, 200000000);

Puede agregar más datos si lo requiere para validar el rendimiento del servicio cuando existan muchas campañas.

> Ejemplo
>
> Por ejemplo, supongamos que queremos ordenar las campañas que mayor recursos recaudó (amount), el resultado esperado sería **RobinFood 2.0,Smoking Burgers,T4 Tea For U.**

Documentación

Se debe entregar la documentación necesaria para ejecutar el *challenge* de manera fluida y entender el proyecto. Cuánta documentación es necesaria queda a criterio del candidato.

Testing

No son necesarios para el desafío. Si se conoce un framework de testing, implementar tests unitarios simples para demostrar el nivel de conocimiento del framework.

Documentación de ayuda para los diferentes lenguajes:

Java:

https://www.youtube.com/watch?v=97qwPG84n7U

https://www.youtube.com/watch?v=aLUHoorbeQk

https://spring.io/projects/spring-boot

Python:
https://www.youtube.com/watch?v=Esdj9wlBOaI&t=727s

https://www.youtube.com/watch?v=IgCfZkR8wME

Docker:

https://www.youtube.com/watch?v=2zlP6n7kQ68

https://www.youtube.com/watch?v=GEiSRdAFXCU&t=31s

Postman

https://www.youtube.com/watch?v=_kX9vKOf0tA&list=PLeo6Q1inqlOeC_zQMg2i3aZcGF_Jmyqd4

En caso de tener dudas con el enunciado o alcance se pueden comunicar enviando sus dudas al correo: paola.avella@bvc.com.co