



Prueba técnica - Jr. Data Scientist

La prueba técnica para Jr. Data Scientist tendrá por objetivos:

- Evaluar las capacidades técnicas del candidato en su uso de herramientas para el análisis de datos e información.
- Evaluar el tren de pensamiento del candidato para resolver un requerimiento presentado de forma ambigua.
- Evaluar la capacidad técnica para descargar información de un REST API.
- Evaluar la capacidad técnica para crear un repositorio Git que permita versionar el código.
- Evaluar la capacidad del candidato para analizar, depurar y agregar información.
- Evaluar la capacidad del candidato para resumir la información en una visualización que permita sacar conclusiones.
- Evaluar el código del candidato y su apego a las mejores prácticas de programación en un ambiente de ciencia de datos.

El problema a resolver será:

¿Cuáles son los 5 artistas más prometedores del Q3 2021?

Información relevante a conocer:

- El candidato deberá hacer uso de toda la información que pueda descargar y analizar del [API de Spotify](#).
- El candidato deberá subir su código a un **repositorio Git** que podamos acceder para revisar el código (la herramienta de versionado quedará a discreción del candidato).
- No es necesario que el código contenga una aplicación final o interactiva, pero sí es importante que el código contenga todos los pasos necesarios para llegar a las conclusiones de datos que alcancen (descarga, depuración, filtrado, visualización).
- El caso deberá presentarse en un **formato Word** o similar donde el candidato deberá escribir todo su tren de pensamiento para llegar a la conclusión que presente.
- Casos presentados en **formato PowerPoint o similar restarán puntos**.
- Casos con **faltas de ortografía restará puntos**.
- En ese formato Word, deberán incluirse los plots o visualizaciones que soporten la lógica del candidato, no se aceptarán conclusiones basadas en gustos personales o conclusiones que no estén soportadas por datos.
- Únicamente se aceptarán soluciones presentadas en los lenguajes de programación R o Python, utiliza con el que te sientas más cómodo y agrega todas las librerías y paqueterías que consideres necesarias, pero no utilices ningún otro lenguaje.

Tips y consejos:

- Para nosotros es más importante evaluar **cómo piensas** que la conclusión a la que llegues sea la correcta.
- Lee bien el problema antes de sacar conclusiones, cualquier lógica que no se adhiera al problema que se plantea restará puntos (cuidado con las fechas).
- Toda **conclusión debe estar soportada por datos** y no deben realizarse saltos lógicos, asunciones o supuestos que no estén soportados por datos.
- El problema está planteado de forma ambigua (¿qué significa “prometedores”?). Esto no se trata de un error, el primer paso del caso se trata en responder a esa pregunta de la mejor forma posible.
- Aunque la presentación sea en formato Word, es **importante adjuntar visualizaciones para resumir los datos**, debe entenderse como la presentación de una solución a un problema planteado por un equipo directivo.
- El equipo que evaluará el Word será diferente al equipo al que evaluará el código en el repositorio Git, mantén contentos a ambos.
- En el código es importante seguir las **mejores prácticas de programación** y evitar código en espagueti, nombres de variables confusos o flujos extraños del código.
- Si en algún punto tienes dudas de cómo resolver el problema, incluye en el Word con qué duda te encontraste y cómo la resolviste (show your work!).