

Prueba Técnica Ingeniero de Software

En **Go Bravo** en el día a día interactuamos con varios sistemas web, proveedores y servicios de otros equipos. Algunos de los componentes que creamos y usamos son Apis, sistemas Web, servicios, además de crear y modificar tablas en PostgreSQL.

Problema a resolver

Desarrollar un sistema para consultar el clima de una ciudad de cualquier parte del mundo, el sistema debe ser capaz de mostrar los detalles de las ciudades que se agreguen como favoritas y debe cumplir con los siguientes requerimientos:

Requerimientos

Consulta de ciudad

- Debe tener un buscador para consultar cualquier ciudad del mundo.
- Debe mostrar los resultados de las coincidencias de las ciudades con el texto ingresado en el buscador.
- Debe tener una opción para poder agregar a favoritos la ciudad deseada.

Consultar el clima de una ciudad

- Al seleccionar una ciudad de la lista de favoritos mostrar el clima.
- Debe mostrar el clima actual en grados centígrados, la temperatura mínima y máxima.
- Mostrar la temperatura por hora para las próximas 24 horas.
- Mostrar el clima mínimo y máximo por día por el resto de la semana.

Consideraciones

- En **Go Bravo** preferimos y usamos **Elixir** pero la tecnología y herramientas a usar queda a decisión del postulante, se recomienda usar el stack que domine mejor.
- La forma de desarrollar el sistema la define el postulante, puede ser web, un cli, una interfaz etc.
- Las funcionalidades antes mencionadas son las mínimas requeridas sin embargo si el postulante ve necesario agregar funcionalidades extra para una mejor usabilidad sumará puntos.
- La forma de entrega es en un repositorio de github.

- Readme
 - Describe el principal problema que resolviste y la solución
 - Explicación de la arquitectura que elegiste
 - Trade-offs que ves en tu implementación y si tuvieras más tiempo qué cambios o cosas harías diferente

Qué se va a revisar

- **Arquitectura:** ¿La solución permite una fácil extensión o modificación de funcionalidades?
- **Exactitud:** ¿Se realizó lo que se pide en la prueba?
- **Calidad del código:** ¿El código es simple, fácil de entender y mantenible?
- **Seguridad:** ¿Hay fallos de seguridad evidentes?
- **Testing:** ¿Tiene pruebas automatizadas?
- **UX:** ¿Es entendible la interfaz y fácil de usar?
- **Decisiones técnicas:** ¿Hacen sentido las librerías, arquitectura que usaste para la aplicación?
- **Escalabilidad:** ¿Va a escalar bien cuando crezca el número de usuarios?
- **Listo para producción:** ¿Cumple con todos los requisitos para ir a producción?

Notas

- Estimamos un tiempo de desarrollo de dos días, pero puedes tardarte lo que requieras.
- No te preocupes tanto por la interfaz de visualización. Lo que nos importa es ver cómo realizaste el problema.
- La Api propuesta para consumir información es la siguiente:
<https://openweathermap.org/>