

# CHALLENGE Intuit

Para desarrolladores

## Pasos

1. **Talento Humano** envía este texto por mail con copia a los **Referentes Técnicos** del equipo para que en la misma cadena se puedan canalizar las consultas.
2. Una vez entregado el enunciado, el mismo día o el siguiente deberías:
  - a. Definir cuál o cuales consignas vas a realizar
  - b. Enviar una estimación de tiempos de realización.
  - c. La idea es que este challenge no lleve más que unos pocos días. Obviamente, se tiene en consideración distintas situaciones personales y laborales que puedan alterar los tiempos previstos.
3. Se inicia el challenge
  - a. Se desarrolla
  - b. Se consultan las dudas con los RTs
  - c. Se entrega el resultado.

## Forma de entrega

- Los fuentes en una carpeta compartida, ZIP, email, etc.
- Durante la entrevista, conversaremos sobre el código, pero idealmente la aplicación debería poder correr en algún entorno. Puede ser en tu propia máquina, compartiendo pantalla. También puede estar subida a algún entorno de pruebas, o lo que creas conveniente.

## Puntos a evaluar

- Porcentaje de concreción de los puntos
  - Si no se llega a completar 100%, el trabajo sirve igual! Porque lo que se busca es analizar y ponderar tus conocimientos y habilidades y dar un feedback constructivo, no desarrollar un producto terminado.
- Calidad del código
  - Limpieza, claridad, prolijidad
  - Funcionamiento/performance
- Comunicación durante el proceso
  - Claridad y pertinencia de las consultas realizadas a los RTs
- Comunicación a la hora de la entrevista
  - Claridad a la hora de mostrar y explicar los puntos relevantes o interesantes de la app. Poder responder las preguntas que se te hagan
- Si existe la posibilidad, realizar alguna pequeña modificación a la app durante la entrevista.

# Consignas

## Backend

Básicamente la idea es crear una API para mantener una tabla de clientes (ABM de Clientes).

Los campos son los siguientes:

- ID (numérico)
- Nombres (alfanumérico)
- Apellidos (alfanumérico)
- Fecha de nacimiento (fecha)
- CUIT (alfanumérico)
- domicilio (alfanumérico)
- teléfono celular (alfanumérico)
- email (alfanumérico)

Para esto se pide que crees una tabla en una base de datos. Los nombres exactos de las columnas serán definidas por vos mismo, al igual que los parámetros de entrada y salida de los métodos de la API.

Se puede usar cualquier motor de base de datos relacional: MS SQL Server, MySQL, PostgreSQL, etc...

Se tiene que desarrollar la API en el estilo API REST, la cual debe correr en un servidor HTTP. Puede ser en cualquier versión de .Net. o bien en PHP, NodeJS, etc.

Los puntos a desarrollar son los siguientes:

1. Crear la tabla y cargar algunos datos de prueba
2. Crear la API y resolver la conexión a la base de datos
3. Implementar los siguientes métodos:
  - a. GetAll. Obtiene todos los registros de la tabla.
  - b. Get (ID). Obtiene un registro correspondiente al ID
  - c. Search. Búsqueda por nombre (caracteres centrales)
  - d. Insert. Crea un registro nuevo
  - e. Update. Actualiza un registro

Los endpoints se ejecutarán desde un cliente tipo PostMan, Insomnia o similar.

Extras:

- Validar la unicidad del campo ID
- Validar los datos
  - Nombres, apellidos, CUIT, teléfono celular, email -> que sean obligatorios
  - Que Fecha de nacimiento, Email y CUIT estén correctamente formateados
- Cualquier mejora propuesta siempre es bienvenida y suma a la evaluación de este challenge.

Ejemplos:

- Registrar un log de errores que se produzcan
- Realizar alguna documentación detallando el funcionamiento de la API



## Frontend

Se trata de crear una App que muestre los datos del clima utilizando la API del sitio web MetaWeather.

La URL con la documentación de la API está en <https://www.metaweather.com/api/>

Puntos a desarrollar:

- Buscador de ciudades. Se debe implementar un buscador en forma de input de texto y mostrar los resultados. O bien usar un campo tipo “auto-complete”, un combo o lo que el candidato crea conveniente.
- Una vez seleccionada una ciudad, consultar los siguientes datos del día actual:
  - Temperatura máxima y mínima
  - Humedad
  - Probabilidad de lluvias (predictability)
- Mostrar los datos de los próximos 5 días

Extras:

- La API soporta búsquedas históricas. Implementar un selector de fechas y efectuar esa búsqueda y mostrar los resultados
  - Tomar solamente el primer resultado obtenido (vienen varias predicciones con cada request) y mostrar en pantalla los mismos datos del punto anterior.
- Como mejora, guardar en caché las consultas realizadas y devolverlas en caso de que ya existan. La duración del caché es de 2 minutos (para simular su funcionamiento). Nota: no se apunta a la utilización de la API de caché de los service workers, si no a implementar a mano una solución sencilla.

Se pueden utilizar los frameworks que el candidato elija y considere convenientes. Tanto los que falicitan la componentización (Angular, VueJS, React) como los que implementan componentes visuales.

## Consignas Fullstack

### Opciones para realizar el challenge fullstack

- El candidato podría desarrollar los dos challenges (el visor del clima en frontend y la API de clientes em backend) por separado
- O bien podría realizar el challenge de backend y desarrollar un frontend sencillo para operar ese ABM de clientes
- Entendiendo posibles limitaciones de tiempo, otra opción podría ser seleccionar uno de los dos challenges y responder preguntas acerca del otro.