

# Desafío Técnico

## Consigna

El desafío consiste en desarrollar un modelo predictivo que permita inferir el revenue generado por un usuario, que acaba de llegar a un juego, dentro de los primeros 7 días desde el momento de la instalación.

Dicho modelo debe ser disponibilizado mediante un microservicio que exponga un endpoint al cual se consulte y devuelva el revenue inferido. Tener en consideración que el sistema se espera que sea utilizado en *real-time* por lo que la velocidad de respuesta del mismo es un factor importante.

## Consideraciones

### General

- En caso de incertidumbre en alguno de los puntos, la definición queda a criterio del desarrollador.
- Se espera que el desarrollo del modelo predictivo, con todas sus etapas, se realice en un notebook.
- Se espera que el tiempo invertido en desarrollar el modelo y el microservicio sea similar.

### Modelo predictivo

- Utilizar preferentemente Python. Las librerías a emplear quedan a criterio del desarrollador.
- El método de evaluación de performance del modelo queda a criterio del desarrollador.

### Microservicio

- Utilizar preferentemente Flask.

### Documentación

- La documentación a entregar, en cantidad y formato, queda a criterio del desarrollador. Mínimamente se espera encontrar la documentación necesaria para entender el proyecto y realizar un deploy en forma fluida.

**Opcionales** (No obligatorios, pero sumar puntos. Los ítems son independientes entre sí).

- Implementar el microservicio mediante Docker.
- Testing. Si se conoce un framework de testing, implementar unit-tests simples para demostrar el nivel de conocimiento del framework.
- Gestionar modelo llevado a producción (microservicio) mediante [MLFlow](#).

- Levantar una base de datos relacional mediante una imagen de docker. Por cada requests hecho a la API dejar un registro en la misma con los datos de entrada y el valor inferido devuelto.

## Datos

El dataset en cuestión cuenta de los siguientes campos:

user\_id: identificador unívoco anonimizado del usuario.

country: país del usuario.

country\_region: sería el equivalente a provincia.

source: si el usuarios llegó en forma orgánica o no-orgánica.

platform: plataforma del usuario.

device\_family: modelo de dispositivo que posee un usuario.

os\_version: versión del sistema operativo.

event\_1: cantidad de eventos de un tipo anonimizado.

event\_2: cantidad de eventos de un tipo anonimizado.

event\_3: cantidad de eventos de un tipo anonimizado.

revenue: revenue generado por el usuario dentro de los primeros 7 días desde la instalación; es la variable a predecir.

## ¿Qué se va a evaluar?

Mediante este desafío buscamos evaluar los siguientes aspectos:

### Machine learning

Buscamos ver conocimiento sólidos en desarrollo de modelos predictivos.

### Desarrollo

1. Estructuración de un proyecto.
2. Buenas prácticas de desarrollo. Modularidad y Flexibilidad. Evitar código *spaghetti*.
3. Uso eficiente de recursos.
4. Criterio para selección de herramientas y bibliotecas.
5. Documentación.
  - a. Ser claro al escribir y comunicar.
  - b. Poder utilizarla para levantar el proyecto sin sufrir.
  - c. Capacidad de ayudar a entender y dar mantenimiento al sistema.

## **Implementación en un entorno productivo**

Durante la devolución podría llegar a haber preguntar sobre cómo sería una posible implementación del sistema en un entorno real.

## **Visión de negocio**

Durante la devolución podría llegar a haber preguntas relacionadas a cómo vincular un desarrollo/modelo con el negocio de la empresa.

# Entrega

Se espera que el proyecto sea trabajado en un repositorio git (gitlab o github). Al momento de entregar el ejercicio:

- Se envíe la URL del proyecto para poder clonarlo y levantarlo en la computadora de la persona que lo revise;
- o en caso de que el candidato no desee tenerlo en un repo público, aceptamos que no envíe un zip del proyecto (tal cual lo que veríamos si hacemos un *git clone* del mismo).