

Ejercicio de entrevista técnica - Software Engineer Back Office

Instrucciones

Esta entrevista es para ver cómo programas y tu capacidad para interpretar una tarea y conceptualizarla. Puedes utilizar todos los recursos que necesites: Google, tus notas, etc. Si te atorras no te preocupes, es normal que las cosas salgan mal al programar, intenta resolverlo.

El ejercicio que vas a hacer está diseñado para ser algo cercano a lo que harías en un día de desarrollo normal. Lo más importante del ejercicio es cumplir con el success criteria que se te pone, fuera de eso hay mucha libertad y flexibilidad para hacer las cosas que se necesiten sin tener que complicarnos de más.

En Atrato utilizamos React para el front end y Node.js junto con Express para nuestro servidor. No es necesario ser un experto en estas tecnologías, pero sí que podamos utilizarlas según sea necesario.

Para hacerte más sencillo el ejercicio creamos un repositorio en el cuál simplemente utilizamos Create React App para hacer la configuración del front end y te creamos el primer archivo del backend que es `server.js`, el cual contiene el código mínimo para hacer funcionar servidor con express y una conexión a la base de datos lo más sencillo posible. Además de eso te creamos dos comandos para que te sea más sencillo programar:

- `npm start` → Default de create react app
- `npm run backend` → El cual corre el backend con nodemon para que no tengas que estar corriendo las cosas cada vez que hagas un cambio

Para clonar el repositorio puedes hacerlo desde este link de Github:

<https://github.com/Atrato-Pago/technical-interview>

Indicaciones para la conexión de la base de datos

Tenemos que tener instalado MySQL, para lograr esto vamos a instalar el servidor y workbench. Aquí el proceso cambia dependiendo del sistema operativo pero en

realidad no es tan relevante.

1. Instalar software

Vas a instalar 2 cosas diferentes:

- MySQL server - entrando a esta página: <https://dev.mysql.com/downloads/mysql/>
- MySQL Workbench - entrando a esta página: <https://dev.mysql.com/downloads/workbench/>

2. Configuración workbench

Una vez instalado workbench que crear una base de datos de manera local para poder conectarla con el servidor. Para esto sigue las siguientes instrucciones:

1. Abre MySQL Workbench
2. Crear un usuario llamado `atrater` en (Mysql > Local Connection > Administration > User and Privileges) con contraseña `Atr4t02021`
3. Dale todos los permisos de `Schema Privileges`
4. Crear la base de datos `atrato`

Teniendo esto listo podemos comenzar a desarrollar el ejercicio

Challenge

Contexto

Somos una empresa que maneja distintas suscripciones a plataformas: Netflix, PrimeVideo y Disney+.

Nuestra empresa recibe pagos de nuestros clientes según las suscripciones que tenga cada cliente. Nuestros clientes pueden pagar ya sea quincenal o mensual y solo pueden pagar los días 1 o 15 de cada mes, o el siguiente día hábil según corresponda.

Queremos desarrollar una plataforma para poder visualizar de alguna manera el siguiente pago de nuestros clientes junto con información relevante, tomando en cuenta qué suscripciones tiene activas para saber el monto del pago, y dependiendo de si es mensual o quincenal la fecha de siguiente pago y con esto también poder hacer algún tipo de proyección de ingresos a futuro.

¿Qué esperamos?

El ejercicio consiste en resolver las siguientes tareas:

- Poder visualizar el siguiente pago de cada cliente
 - Datos a visualizar: nombre, datos de contacto, cantidad de suscripciones contratadas, total a pagar en su siguiente pago, fecha de ultimo pago, fecha del siguiente pago, periodicidad de pago (mensual o quincenal)
 - Poder ver dentro de esta misma visualización, qué suscripciones tiene el cliente

Es necesario generar por lo menos las siguientes 4 tablas en la base de datos y hacer el vaciado de la información:

- clientes (basarse en csv de "users.csv")
- suscripciones (basarse en csv de "subscriptions-users.csv")
- pagosRecibidos (basarse en csv de "paymentsReceived.csv")
- plataformas (basarse en csv de "platforms.csv")

A estas tablas hay que guardarles la información que se encuentra en los archivos CSV de la sección **Material necesario**.

Forzosamente la tabla de pagosRecibidos sólo tendrá los campos de "id", "dateCreated", "amount" y "userId", todas las demás tablas pueden tener las columnas que sean necesarias.

Siéntete con la libertad de agregar más tablas y/o columnas a la base de datos si es que lo ves necesario

Los pagos esperados se deben calcular con base en el último pago recibido de cada cliente, considerando la periodicidad de pago (mensual o quincenal). Ejemplo: Si un usuario pagó el 15 de enero y tiene la periodicidad mensual, entonces su siguiente pago esperado deberá ser el 15 de febrero. Si este usuario tuviera la periodicidad como quincenal, entonces su siguiente pago debería de ser el 1 de febrero.

Toda la data se generó con información "dummy", entonces puede haber casos en los que la información no tenga "sentido". Puede ser que un cliente tenga 0 suscripciones, pero sí tenga pagos, o todo lo contrario, que un cliente tenga 5 suscripciones, pero ningún pago. Vamos a asumir que todo tiene sentido, pero únicamente en los casos en los que sí se tengan suscripciones y no tenemos información para calcular el siguiente

pago (porque no hay pagos registrados) entonces se deberá tomar la fecha de siguiente pago como la siguiente fecha válida disponible (días 1 o 15 y que no sea fin de semana o día feriado)

Success criteria

- ☐ Crear API para poder obtener la información a visualizar en el Front End a través de la conexión a la base de datos existente en el servidor
 - ☐ Manejo de "Loading" y "Errores" correcto
 - ☐ No dejar al usuario nunca con una "pantalla blanca"
- ☐ Crear interfaz para visualizar información de los siguientes pagos
 - ☐ No se esperan pagos en fechas inválidas (fines de semana o días feriados)
 - ☐ Las fechas de pago esperadas son en los días 1 o 15, según corresponda, o el siguiente día hábil
- ☐ Obtener la información a visualizar desde la base de datos local
- ☐ Generar tablas y guardar información correctamente
- ☐ Poder explicar detalladamente la solución realizada

Material necesario

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/e8756f80-d7ff-4303-803e-6c5e424b578a/diasFeriados.csv>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/473315ca-f694-44b7-9a34-0393e4b9ca72/paymentsReceived.csv>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d2cab0dc-724b-45ae-8aa7-0dad9e372e6e/suscriptions-users.csv>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/653d6299-d794-456b-a83c-a78176452141/users.csv>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/6ffdde76-5fc1-46d7-9b5e-884166afa683/platforms.csv>

Como nota adicional, los pagos recibidos comienzan desde enero del 2021 y no hay data más allá del primero de noviembre de 2021