

Challenge - Truora

Antes de iniciar recuerda

Si vas a poner el código en un repositorio público, **no uses el nombre de Truora en ninguna parte.**

Este es un **reto de aprendizaje** y no esperamos que conozcas las tecnologías necesarias para resolverla. Esperamos que puedas aprender y aplicar.

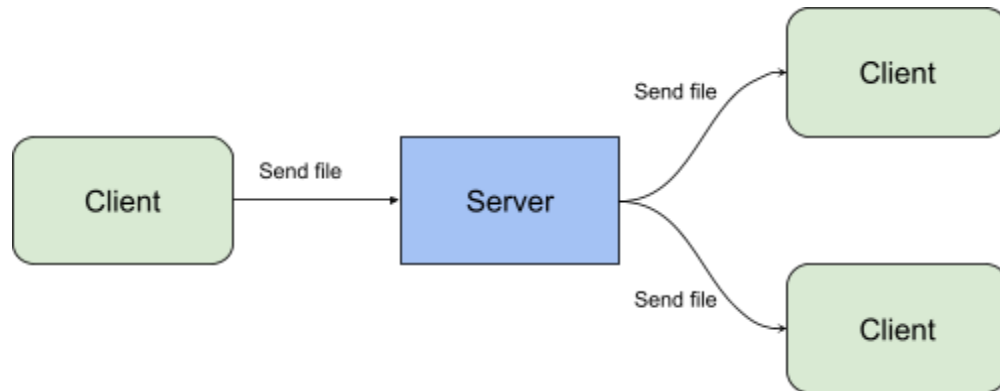
Challenge - Truora	1
Definición del problema	2
Parte 1: Diseñar el protocolo	2
Parte 2: Escribir el servidor	3
Parte 3: Escribir el cliente	3
Parte 4: (Opcional) Escribir una interfaz web para el servidor	3
Tecnologías	3
Notas	4

Es normal en Truora enfrentarse a desafíos de los que no sabemos mucho o a veces nada. Esta prueba tiene como objetivo presentar un reto similar en el que los ingenieros tienen que investigar una solución e implementarla en un periodo corto de tiempo.

Ésta prueba también te sirve cómo candidato para saber si es un trabajo que te va a gustar o no.

Definición del problema

Crear un servidor que permite transferir archivos entre 2 o más clientes usando un *custom protocol* (protocolo no estándar) basado en TCP.



El UX es cliente y servidor es libre pero una posible solución sería:

Iniciar servidor:

```
$ ./server start
```

Poner 3 clientes en modo recibir archivo:

```
$ ./client receive -channel 1
```

```
$ ./client receive -channel 1
```

```
$ ./client receive -channel 2
```

Enviar archivo:

```
$ ./client send miarchivo.png -channel 1
```

```
$ ./client send algo.txt -channel 2
```

El cliente también puede ser una aplicación para celular.

Parte 1: Diseñar el protocolo

Diseña un protocolo simple de comunicación entre el cliente y el servidor, el objetivo es recibir y enviar archivos.

El protocolo debe permitir al cliente suscribirse a canales y también enviar archivos a canales específicos.

Un ejemplo de un protocolo sencillo podría ser:

Cliente suscribiéndose a un canal:

```
subscribe {chan}
```

Cliente enviando un archivo:

```
send {chan} {content-size}  
{datos}
```

Parte 2: Escribir el servidor

El servidor es una herramienta de línea de comandos que coordina a los clientes

Parte 3: Escribir el cliente

El cliente es una herramienta de línea de comandos o una aplicación para celular que permite enviar y recibir archivos por canales

Parte 4: (Opcional) Escribir una interfaz web para el servidor

Esta interfaz sería para el servidor, en ella debería mostrar qué clientes están conectados, en qué canales y estadísticas de envío de los archivos.

Tecnologías

Las siguientes tecnologías **tienen** que ser usadas para resolver la prueba. Entendemos que es muy posible que no tengas experiencia usando estas tecnologías. Para nosotros, es muy importante medir la capacidad de aprendizaje que tiene un/a ingeniero/a. Esto además, ayuda a entender de mejor forma el talento.

Lenguaje servidor: Go

Cliente: Línea de comandos (Go) o iOS (Swift) o Android (Java o Kotlin)

Interfaz servidor: Vue.js

Si usas Windows, te recomendamos instalar
<https://docs.microsoft.com/en-us/windows/wsl/install>

Notas

- El diseño de UI/UX es libre si decides hacer esta parte
- La prueba debe ser realizada por una sola persona y sustentada en vivo
- El nivel (IC1..IC6) es determinado por el resultado de la prueba respecto a unos criterios bien definidos
- No buscamos una solución absolutamente perfecta, buscamos la solución de cada persona respecto a su nivel. Si tu nivel es de Senior esperamos ver una solución de nivel Senior.
- Limite de tiempo: 4 semanas
- Se puede seguir mejorando la prueba hasta el día de la sustentación
- Durante la entrevista vamos a revisar lo que hiciste corriendo en tu computador
- Si usas Windows, te recomendamos instalar <https://docs.microsoft.com/en-us/windows/wsl/install>

Recuerda

Si vas a tomar... no manejes.

Y si vas a poner el código en un repositorio público, **no uses el nombre de Truora en ninguna parte.**