# 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.

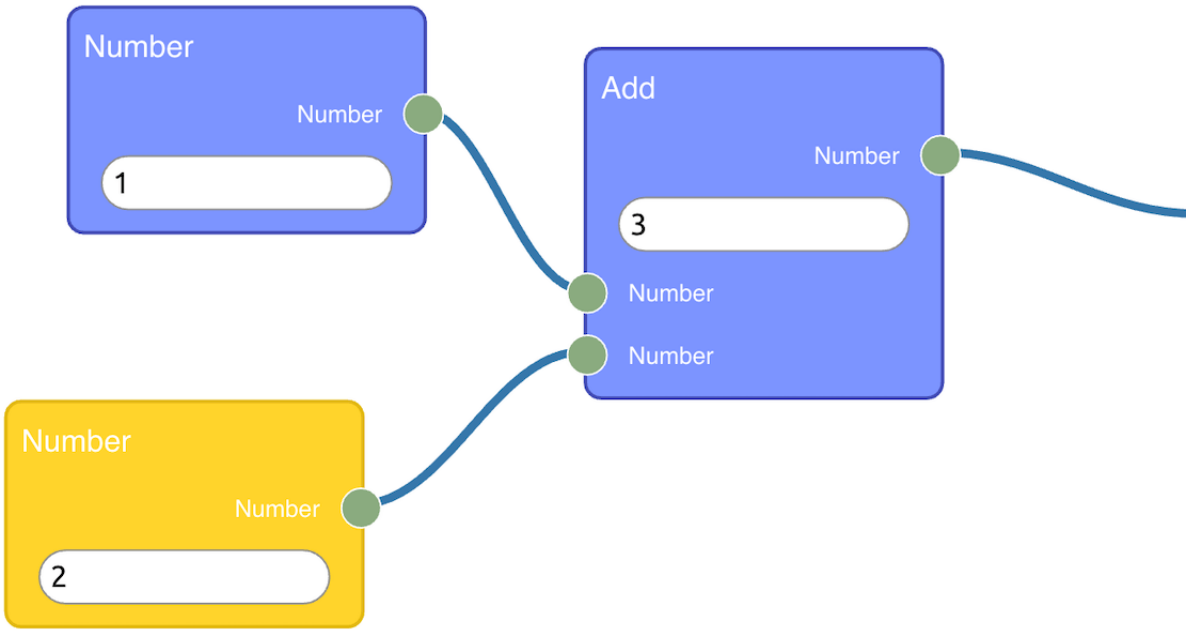


Es normal en Truora enfrentarse a desafíos de los que no sabemos mucho o a veces nada. Esta prueba tiene como objetivo presentar un reto similar en el que los ingenieros tienen que investigar una solución e implementarla en un periodo corto de tiempo.
Ésta prueba también te sirve cómo candidato para saber si es un trabajo que te va a gustar o no.

## Definición del problema

Crear herramienta para aprender a programar visualmente
La herramienta debe permitir crear **programas muy simples** usando un editor visual de nodos.

Por ejemplo, crear un programa que sume los números 1 y 2 se haría creando un nodo "**Add**" que tiene como entradas los números 1 y 2, por ejemplo:




En este caso, por ejemplo, la salida de "**Add**" se podría asociar a un nodo "**Assign**" que asignaría su valor.

Como mínimo, el editor debe soportar:

- Bloques de código
- Asignación
- Operaciones matemáticas básicas
- Estructura de control If-Else
- Estructura de control For básica (desde-hasta)

El editor debe finalmente general código Python y debe poder evaluarlo

De una manera simple, el editor permite crear ASTs simples, convertirlos a código y evaluarlos.

## Parte 1: Crear Editor de Nodos

El editor debe ser construido con el framework Vue.JS y el editor con DrawFlow (https://github.com/jerosoler/Drawflow).

Debe permitir programar usando el teclado y el ratón, también debe permitir guardar y cargar programas desde el backend

### Parte 2: Visualización del código

Los nodos creados deben convertirse a código Python y mostrarse en la interfaz. Debe permitirse ejecutar el código y mostrar los resultados.

### Parte 3: Backend

El backend debe ser un REST API compuesto de endpoints para:

- Guardar el programa
- Listar programas guardados
- Obtener el programa
- Ejecutar el programa

### Parte 4: Opcional

Si te animas, agregale soporte para otros lenguajes de programación.
Otras ideas serían otros tipos de datos como “string”, y funcionalidades como hacer peticiones HTTP

## Tecnologias

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 Backend:** Go
**Base de Datos:** Dgraph
**API Router:** chi (pronunciado kai)
**Interfaz:** Vue.js DrawFlow https://github.com/jerosoler/Drawflow

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

## Notas

- El API puede ser REST
- El diseño de UI/UX es libre
- 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 (tiempo promedio: 1 semana)
- Se puede seguir mejorando la prueba hasta el dia 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.**

-