

# 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.

Punto 1: Protocolos	1
<b>Punto 2: Arquitectura</b>	<b>2</b>
Punto 3: Implementación	2
Operación Index Select	2
<b>Notas Generales</b>	<b>3</b>

## Punto 1: Protocolos

Diseñar protocolo para intercambio de archivos encriptados entre un SDK para móviles y un backend HTTPS

Notas:

- El protocolo debe funcionar como una capa adicional a TLS
- Esperamos precisión en los algoritmos y sus con parámetros de configuración
- Esperamos diagramas que ilustran el protocolo
- En este punto no esperamos ninguna implementación
- Puede ser documentado en un Google doc o herramienta similar

## Punto 2: Arquitectura

Diseñar un API para una empresa B2B que recibe credenciales del banco Truvivienda (entidad externa) de un usuario.

El API consulta los extractos bancarios y créditos del usuario y además consulta la página de información de aportes sociales para finalmente obtener una calificación crediticia.

Notas:

- Esperamos diagramas que ilustren el diseño de la arquitectura
- En este punto no esperamos ninguna implementación
- Puede ser documentado en un Google doc o herramienta similar

## Punto 3: Implementación

Implementar estructura de datos [Tensor](#) y operaciones Reshape, Hadamard Product e Index Select

### Operación Index Select

Recibe un tensor, la dimensión de la cual tomar los datos y un vector de índices, ejemplos:

```
=> IndexSelect([1, 2, 3, 4], 0, [0, 0, 2])  
[1, 1, 3]
```

```
=> IndexSelect([[1, 2], [3, 4]], 0, [0])  
[[1, 2]]
```

```
=> IndexSelect([[1, 2], [3, 4]], 0, [0, 0])  
[[1, 2], [1, 2]]
```

```
=> IndexSelect([[1, 2], [3, 4]], 0, [0, 0, 1, 1])  
[[1, 2], [1, 2], [3, 4], [3, 4]]
```

```
=> IndexSelect([[1, 2], [3, 4]], 1, [0])  
[[1], [3]]
```

```
=> IndexSelect([[1, 2], [3, 4]], 1, [0, 0])
```

[[1, 1], [3, 3]]

=> IndexSelect([[1, 2], [3, 4]], 1, [0, 0, 1, 1])

[[1, 1, 2, 2], [3, 3, 4, 4]]

Notas:

- Esperamos una implementación simple de la estructura de datos
- Implementar en Go aunque no conozcas el lenguaje de programación
- La implementación no debe usar librerías externas
- La implementación sólo debe funcionar para el tipo de dato float32
- Puede ser subido a Github, Gitlab, Bitbucket o similares pero no uses ni el nombre tensor ni el de Truora en el nombre del proyecto ni en el README

## Notas Generales

- En esta prueba a diferencia de las pruebas para todos los niveles es posible que no demos ningún feedback
- Puedes aplicar a esta misma posición usando las otras pruebas para todos los niveles
- El tiempo máximo para entregar la prueba son: 2 semanas