



# Transfiriendo

Hazlo todo más fácil

## Prueba de conocimiento desarrollador junior

### 1) Conocimiento JavaScript

Instrucciones: Crea una función en JavaScript que, dada una lista de enteros, devuelva el número más grande posible que se puede formar concatenando los números.

Requisitos:

La función debe recibir un arreglo de números enteros como entrada.

La salida debe ser el número más grande posible formado al concatenar estos números como una cadena.

Ejemplo:

Input: [3, 30, 34, 5, 9]

Output: "9534330"

### 2) Conocimiento Node Js

Instrucciones: Crea una API REST simple usando Node.js y Express que permita gestionar una lista de tareas (ToDo list).

Requisitos:

La API debe permitir las siguientes operaciones:

Crear una nueva tarea.

Obtener la lista de todas las tareas.

Actualizar una tarea (marcarla como completada o modificar el contenido).

Eliminar una tarea.

Cada tarea debe tener la siguiente estructura:

```
json
{"id": Number,
"title": String,
"completed": Boolean}
```

No es necesario conectar la API a una base de datos; puedes manejar las tareas en memoria (un arreglo en el servidor).

+601 6910810

info@transfiriendo.com

Av. Carrera 45 N° 108-27 Torre 2 Piso 17 Paralelo 108, Bogotá

Presencia en: Colombia  Ecuador  Argentina 

[www.transfiriendo.com](http://www.transfiriendo.com)





## Rutas esperadas:

POST /tasks para crear una nueva tarea.  
GET /tasks para obtener todas las tareas.  
PUT /tasks/:id para actualizar una tarea.  
DELETE /tasks/:id para eliminar una tarea.

## 3) Conocimiento React

Instrucciones: Crea una pequeña aplicación de React que consuma la API creada en la Parte 2 para mostrar una lista de tareas y permitir al usuario interactuar con ella.

## Requisitos:

La aplicación debe mostrar todas las tareas obtenidas desde la API.

El usuario debe poder:

Crear una nueva tarea desde un formulario.

Marcar una tarea como completada.

Eliminar una tarea de la lista.

La interfaz debe ser sencilla pero funcional, utilizando componentes como useState y useEffect para manejar el estado y las peticiones a la API.

Opcional: Usa algún estilo básico con CSS o un framework como Material-UI o Bootstrap para darle una presentación simple y organizada.

## Instrucciones finales:

Subir el código a un repositorio en GitHub.

Proporcionar un archivo README con instrucciones para correr la API y la aplicación de React.

**Nota:** al momento de finalizar la prueba entregar la URL del GitHub

