1. Crear un algoritmo que muestre los números impares entre el 0 y el 100.

const impar = () => {

    let arrImpar = []

    for(let i=0; i<100; i++){

        if( i % 2 !== 0 ) arrImpar.push(i)

    }

    return arrImpar

}

2. Realizar un programa que ingrese los sueldos de 5 operarios en un vector. Realizar la creación y carga del vector en el constructor. Crear un método para imprimir el vector.

cargarSueldos = (*sueldo1*, *sueldo2*, *sueldo3*, *sueldo4*, *sueldo5*) =>{

    let sueldos = **new** *Array*(*sueldo1*, *sueldo2*, *sueldo3*, *sueldo4*, *sueldo5*)

    console.log(sueldos)

}

cargarSueldos(1,2,3,4,5)

3. Plantear una clase llamada Alumno y definir como atributos su nombre y su edad. En el constructor realizar el ingreso de datos. Definir otros dos métodos para imprimir los datos ingresados y un mensaje si es mayor o no de edad (edad >= 18)

class Alumno {

    constructor(*nombre*, *edad*){

*this*.nombre = *nombre*

*this*.edad = *edad*

    }

    getData(){

        console.log(*this*.nombre + ", " + *this*.edad + " años")

    }

    getAge(){

        console.log(*this*.edad >= 18 ? "Mayor de edad" : "Menor de edad")

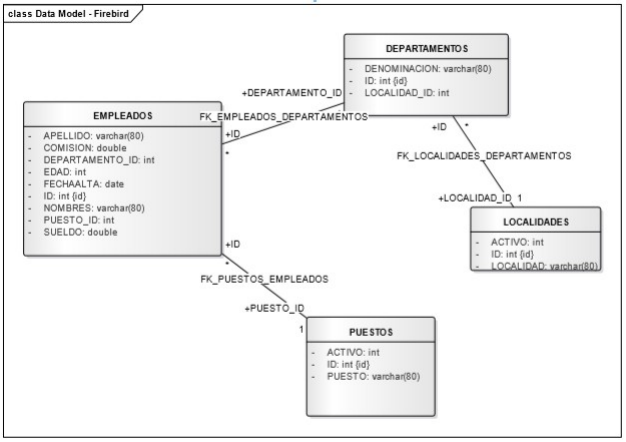
    }

}

4. JavaScript ES6: Dados los siguientes array, imprimir por consola los elementos del array “y” que no se encuentran en el array “x” utilizando para tal fin una única línea de código. const x = ["n", "bro", "c", "|"]; const y = ["d", "n", "l", "bro", "g"];

y.map( *yy* => x.includes(*yy*) === false? console.log(*yy*) : null )

5. Dada la siguiente base de datos relacional: RESUELVA LAS SIGUIENTES CONSULTAS EN SQL:



**ENTIDAD**

**ATRIBUTOS**

1. Mostrar los nombres de los empleados ordenados alfabéticamente (Z...A)   
     
   SELECT nombres FROM empleados ORDER BY nombres DESC;
2. Seleccionar el nombre, el puesto y la localidad donde trabajan los empleados con puesto de ‘Soporte’.

SELECT nombres, puesto, localidad FROM empleados e, departamentos d, localidades l, puestos p WHERE p.puesto='Soporte' AND e.puesto\_id=p.id AND d.id=e.departamento\_id AND d.localidad\_id= l.id;

1. Listar los nombres de los empleados cuyo nombre termine con la letra ‘o’.

SELECT nombres

FROM empleados

WHERE nombres LIKE '%o';

1. Seleccionar el nombre, el puesto y sueldo de los empleados que trabajan en la localidad Carlos Paz.

SELECT nombres, puesto, sueldo FROM empleados e, departamentos d, localidades l, puestos p WHERE l.localidad=’Carlos Paz’ AND e.puesto\_id=p.id AND d.id=e.departamento\_id AND d.localidad\_id= l.id;

1. Seleccionar el nombre, sueldo y localidad donde trabajan de los empleados que tengan un sueldo entre 10000 y 13000.

SELECT nombres, sueldo, localidad FROM empleados e, departamentos d, localidades l WHERE e.sueldo > 10000 AND e.sueldo < 13000 AND d.id=e.departamento\_id AND d.localidad\_id= l.id;

1. Visualizar los departamentos con más de 5 empleados

SELECT COUNT(departamento\_id), denominacion FROM empleados e, departamentos d

WHERE e.departamento\_id = d.id GROUP BY departamento\_id , denominacion

HAVING COUNT(departamento\_id)>5;

1. Nombre de los empleados que trabajan en Córdoba y cuyo puesto sea ‘Analista’ o ‘Programador’.

SELECT nombres FROM empleados e, departamentos d, localidades l, puestos p WHERE l.localidad= ‘Córdoba’ AND p.puesto IN(‘Analista’, ‘Programador’) AND e.puesto\_id=p.id AND d.id=e.departamento\_id AND d.localidad\_id= l.id;

1. Calcula el sueldo medio de todos los empleados

SELECT AVG(sueldo) FROM empleados 🡪 Este me saca el sueldo promedio de todos los empleados

SELECT AVG(sueldo), nombres FROM empleados GROUP BY nombres; 🡪 Este me saca el sueldo promedio de

cada empleando mostrando el nombre

del mismo

1. ¿Cuál es el máximo sueldo de los empleados del departamento 10?

SELECT MAX(sueldo) FROM empleados WHERE departamento\_id = 10;

1. Calcula el sueldo mínimo de los empleados del departamento ‘Soporte’

SELECT MIN(sueldo) FROM empleados, departamentos d WHERE d.denominacion = ‘Soporte’;

1. Para cada puesto obtener la suma de sueldos.

SELECT SUM(sueldos), puesto FROM empleados e, puestos p WHERE e.puesto\_id = p.id

GROUP BY puesto;

6. Según el diseño de figma, plasmar en código, un listado de productos con búsqueda por descripción y código de producto.

Los productos pueden venir de un array o un archivo json.

Simular un tiempo de respuesta de 3 segundos con un setTimeOut

Se valora el manejo de estados entre componentes, uso de hooks y estilos.

Diseño

<https://www.figma.com/file/D3jw2OCgmuxMm3YlRo2DPe/Untitled?node-id=0%3A1&t=xOD5k4nlkDXczbV4-0>

**LISTO**

[**https://challenge-flexxus.vercel.app/**](https://challenge-flexxus.vercel.app/)

[**https://github.com/MFlorO/challenge-flexxus**](https://github.com/MFlorO/challenge-flexxus)