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

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.

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)

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"];

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



* Mostrar los nombres de los empleados ordenados alfabéticamente (Z...A)
* Seleccionar el nombre, el puesto y la localidad donde trabajan los empleados con puesto de ‘Soporte’.
* Listar los nombres de los empleados cuyo nombre termine con la letra ‘o’.
* Seleccionar el nombre, el puesto y sueldo de los empleados que trabajan en la localidad Carlos Paz.
* Seleccionar el nombre, sueldo y localidad donde trabajan de los empleados que tengan un sueldo entre 10000 y 13000.
* Visualizar los departamentos con más de 5 empleados
* Nombre de los empleados que trabajan en Córdoba y cuyo puesto sea ‘Analista’ o ‘Programador’.
* Calcula el sueldo medio de todos los empleados.
* ¿Cuál es el máximo sueldo de los empleados del departamento 10?
* Calcula el sueldo mínimo de los empleados del departamento ‘Soporte’.

* Para cada puesto obtener la suma de sueldos.

6. Prueba Técnica: Desarrollo de API con Cumplimiento de Pautas

Descripción del Desarrollo:

Se requiere la creación de un API que cumpla con las siguientes especificaciones:

CRUD de ARTÍCULOS: Implementar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para artículos, que incluyan un identificador único (PK), nombre, fecha de modificación, marca y estado de activación.

Operaciones Específicas:

GET: Permitir filtrar por coincidencia del nombre, estado de actividad y búsqueda exacta.

DELETE: Desactivar el artículo en lugar de eliminarlo directamente.

UPDATE: Actualizar cualquier campo excepto el identificador único (PK), con la opción de actualizar uno o más campos en una sola operación.

INSERT: Requerir los campos de nombre y marca para la inserción.

Pautas a Cumplir:

Arquitectura Adecuada: Diseñar una arquitectura robusta y escalable que facilite el mantenimiento y la expansión futura del API.

Seguridad en las Rutas: Implementar medidas de seguridad para restringir el acceso a las rutas del API, garantizando la confidencialidad e integridad de los datos.

Validación de Datos: Validar tanto la información recibida del cliente como la enviada al cliente para asegurar la coherencia y consistencia de los datos en todo momento.

Código Limpio y Estructurado: Escribir código claro, legible y bien estructurado siguiendo las mejores prácticas de programación, con el objetivo de mejorar la mantenibilidad y comprensión del sistema.

Conexión a la Base de Datos: Establecer una conexión confiable y eficiente con una base de datos, ya sea relacional o no relacional, para almacenar y recuperar datos de manera adecuada.

Documentación del API: Generar una documentación completa y detallada del API, que incluya información sobre sus endpoints, parámetros, respuestas esperadas y ejemplos de uso, con el fin de facilitar su entendimiento y uso por parte de otros desarrolladores.

Repositorio de Código: Subir todo el código fuente del API, incluyendo la documentación y cualquier otro recurso necesario, a un repositorio de control de versiones (por ejemplo, GitHub), para facilitar la colaboración, revisión y seguimiento del desarrollo del proyecto.