



Técnico en Ciberseguridad

Facilitador: Kevin Feliz Henríquez

Nombre del Módulo: Estructuras de Datos y Algoritmos

Nombre Actividad: Actividad 2

Descripción: En esta actividad, abordaremos las operaciones CRUD en una base de datos (Create, Read, Update, Delete), y pondremos todo esos elementos en práctica con una actividad interactiva que servirá como instrumento de aprendizaje.

Actividad #2

Operaciones Básicas en Bases de Datos

Las operaciones más comunes se resumen en el acrónimo **CRUD**:

- **Create** (Crear): Agregar nuevos datos.
- **Read** (Leer): Recuperar datos existentes.
- **Update** (Actualizar): Modificar datos.
- **Delete** (Eliminar): Borrar datos.

Actividad #2

Ejemplos de Operaciones (SQL)

Estos ejemplos se basan en una tabla llamada `productos`.

Operación	Comando SQL
-----------	-------------

:---	:---
------	------

CREATE	<code>INSERT INTO productos (nombre, precio) VALUES ('Laptop', 1200.00);</code>
---------------	---

READ	<code>SELECT * FROM productos WHERE precio > 1000;</code>
-------------	--

UPDATE	<code>UPDATE productos SET precio = 1250.00 WHERE nombre = 'Laptop';</code>
---------------	---

DELETE	<code>DELETE FROM productos WHERE nombre = 'Laptop';</code>
---------------	---

Actividad #2

Ejemplos de Operaciones (MongoDB)

Estos ejemplos se basan en una colección llamada `productos`.

Operación	Comando MongoDB
-----------	-----------------

:	:
---	---

CREATE	<code>db.productos.insertOne({ nombre: "Laptop", precio: 1200 });</code>
---------------	--

READ	<code>db.productos.find({ precio: { \$gt: 1000 } });</code>
-------------	---

UPDATE	<code>db.productos.updateOne({ nombre: "Laptop" }, { \$set: { precio: 1250 } });</code>
---------------	---

DELETE	<code>db.productos.deleteOne({ nombre: "Laptop" });</code>
---------------	--

Actividad #2

Consideraciones de Seguridad: El Modelo de Cliente-Servidor

Es **críticamente importante** entender que **no se debe conectar un programa de JavaScript en el navegador directamente a una base de datos**.

Exponer las credenciales de la base de datos en el código del cliente es un riesgo de seguridad masivo, permitiendo a cualquier persona acceder y manipular tus datos.

La arquitectura correcta utiliza un **servidor intermediario** (backend) para manejar las conexiones.

1. El **cliente** (HTML/CSS/JS en el navegador) no se conecta a la base de datos.
2. En cambio, el cliente hace una petición a una **API** alojada en el servidor (usando `fetch`).
3. El **servidor** (backend) se conecta a la base de datos de forma segura, realiza la operación y envía una respuesta al cliente.