**BASES DE DATOS**

**LAURA JULIETH GUERRERO ROA N° 1053282200**

**MODELADO DE LOS ARTEFACTOS DEL SOFTWARE.**

**ESTUPIÑAN FINO DEMETRIO MAURICIO**

**SERVICIO NACIONAL DE APRENDIZAJE (SENA)**

**BOGOTA D.C**

**2025**

**3.1 Actividades de Reflexión inicial.**

**Actividad:** Luego de hacer una lectura al texto anterior. Redacte un pequeño texto donde usted con sus palabras de a comprender lo expresado en el texto.

Cada vez la tecnología aumenta más, estamos en contantes cambios y avances, ya todo es digital, estudiar, trabajar, comprar todo lo podemos realizar en cuestión de segundos desde cualquier lugar teniendo acceso a internet, en cada cosa que realizamos se están almacenando y procesando nuestros datos, dejamos una huella digital enorme. Y toda esa información es valiosa siempre y cuando sepamos que hacer con ella. Pero ¿Cómo logramos que todo esto funcione? Pues lo logramos por medio del uso de herramientas como SQL que nos permiten no solo consultar esa información, sino también descubrir cosas nuevas, como automatizar procesos y, lo más importante, mantener nuestros datos seguros y en orden**.** SQL es mucho más que entender y saber de tecnología, es una habilidad estratégica que nos permite transformar datos en conocimiento útil. Es la llave para analizar el mundo de manera más profunda, tomar decisiones basadas en evidencia y abrirnos a un sinfín de posibilidades impulsadas por la información que generamos constantemente.

**3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje. (Argumento Técnico)**

**1. PROCEDIMIENTOS ALMACENADOS (Stored Procedures)**

**¿Qué es?**

Un procedimiento almacenado es un conjunto de instrucciones SQL que se almacena en la base de datos y puede ejecutarse cuando se necesite, con o sin parámetros.

**Estructura básica:**

**CREATE PROCEDURE nombre\_procedimiento(**

**IN param1 tipo,**

**IN param2 tipo**

**)**

**BEGIN**

**-- Instrucciones SQL**

**END;**

**Ventajas:**

* Evita la repetición de código.
* Mejora el rendimiento (menos tráfico entre aplicación y base de datos).
* Mayor seguridad (se puede restringir acceso a los datos directos).
* Facilita el mantenimiento del código.

**¿Cuándo usarlo?**

* Cuando necesitas ejecutar una serie de operaciones repetidamente.
* Para encapsular lógica compleja de negocio.
* Al automatizar tareas como inserciones o actualizaciones masivas.

**2. VISTAS (Views)**

**¿Qué es?**

Una vista es una tabla virtual basada en una consulta SQL. No almacena datos por sí misma, pero permite consultar datos como si fuera una tabla.

**Estructura básica:**

CREATE VIEW nombre\_vista AS

SELECT columna1, columna2

FROM nombre\_tabla

**Ventajas**:

* Simplifica consultas complejas.
* Mejora la seguridad ocultando columnas sensibles.
* Facilita el análisis de datos al presentar solo la información necesaria.

**¿Cuándo usarla?**

* Cuando necesitas mostrar solo ciertas columnas o registros a un usuario.
* Para evitar repetir consultas complejas.
* Al combinar información de varias tablas con una sola vista.

**3. FUNCIONES (Functions)**

**¿Qué es?**

Una función es similar a un procedimiento, pero siempre devuelve un valor. Se utiliza dentro de consultas para devolver resultados calculados.

**Estructura básica:**

CREATE FUNCTION nombre\_funcion(param1 tipo)

RETURNS tipo\_retorno

BEGIN

RETURN resultado;

END;

**Ventajas**:

* Reutilización del código para operaciones frecuentes.
* Puede usarse en consultas SELECT, WHERE, ORDER BY, etc.
* Ayuda a mantener el código limpio y modular.

**¿Cuándo usarla?**

* Para cálculos personalizados dentro de consultas.
* Para encapsular operaciones lógicas repetidas (como calcular edad a partir de fecha de nacimiento).
* Cuando se requiere una salida única a partir de parámetros.

**4. TRIGGERS (Disparadores)**

**¿Qué es?**

Un trigger es un bloque de código que se ejecuta automáticamente en respuesta a un evento en una tabla, como INSERT, UPDATE o DELETE.

**Estructura básica:**

CREATE TRIGGER nombre\_trigger

AFTER INSERT ON nombre\_tabla

FOR EACH ROW

BEGIN

-- Código a ejecutar

END;

**Ventajas:**

* Automatiza validaciones o acciones cuando cambian los datos.
* Ayuda a mantener la integridad de los datos.
* Permite auditoría (guardar cambios en tablas de log, por ejemplo).

**¿Cuándo usarlo?**

* Para auditar cambios (crear log de inserciones o modificaciones).
* Para validar o modificar datos antes de insertarlos.
* Para mantener coherencia entre tablas relacionadas.