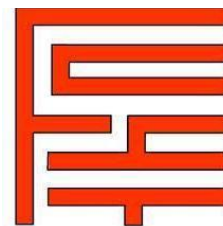


UNIVERSIDAD MAYOR DE SAN SIMÓN
FACULTAD DE CIENCIAS Y TECNOLOGÍA



PROYECTO SEMESTRAL

INFORME 2 - HITO 2

Estudiantes

Garcia Guzman Oliver Saúl
Sarzuri Calizaya Carlos Andrés
Temprano Villarpando Dylan Alex
Vallejos Hidalgo Rody Brayan

Asignatura

Taller de Base de Datos

Docente

Valero Malele Tania Andrea

Grupo

4

Cochabamba – Bolivia

2025-10-15

Índice

1. Introducción	3
2. Triggers	3
3. Base de datos con Datos	5
4. Funciones para CRUD	7
5. Interfaces en HTML con conexión a la base de datos	19
6. Estructura del proyecto	20
7. Cronograma	20

BASH ACADEMY

1. Introducción

En esta parte del segundo hito se irá desglosando y desarrollando diferentes triggers, mockups, el CRUD y el llenado de la base de datos.

A continuación se verá el desarrollo de estos mismos, no será el llenado completo pero sí uno que nos permita ver cómo irá funcionando nuestra base de datos.

El uso correcto de la base de datos será fundamental para que se pueda desarrollar el sistema sin errores.

2. Triggers

1. Reducir cupos cuando un usuario se inscribe

```
CREATE OR REPLACE FUNCTION trg_reducir_cupos()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE curso
    SET cupos = cupos - 1
    WHERE id_curso = NEW.id_curso;
    RETURN NEW;
END;
$$;

CREATE TRIGGER trg_reducir_cupos
AFTER INSERT ON inscripcion
FOR EACH ROW
EXECUTE FUNCTION trg_reducir_cupos();
```

2. Evitar inscripción sin cupos

```
CREATE OR REPLACE FUNCTION trg_verificar_cupos()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
DECLARE
    cupos_restantes INTEGER;
BEGIN
    SELECT cupos INTO cupos_restantes
    FROM curso
    WHERE id_curso = NEW.id_curso;

    IF cupos_restantes <= 0 THEN
        RAISE EXCEPTION 'No hay cupos disponibles';
    END IF;

    RETURN NEW;
```

```

END;
$$;

CREATE TRIGGER trg_verificar_cupos
  BEFORE INSERT ON inscripcion
  FOR EACH ROW
  EXECUTE FUNCTION trg_verificar_cupos();

```

3. Sumar puntos al realizar un pago

```

CREATE OR REPLACE FUNCTION trg_sumar_puntos()
  RETURNS TRIGGER
  LANGUAGE plpgsql
  AS $$
  DECLARE
    puntos_ganados INTEGER;
  BEGIN
    puntos_ganados := NEW.monto * 0.10;

    INSERT INTO puntos (id_usuario, saldo_puntos, motivo, puntos_usados)
    VALUES (NEW.id_usuario, puntos_ganados, 'pago de curso', 0)
    ON CONFLICT (id_usuario)
    DO UPDATE SET saldo_puntos = puntos.saldo_puntos + puntos_ganados;

    RETURN NEW;
  END;
  $$;

CREATE TRIGGER trg_sumar_puntos
  AFTER INSERT ON pago
  FOR EACH ROW
  EXECUTE FUNCTION trg_sumar_puntos();

```

4. Asignar fecha automáticamente a comentarios

```

CREATE OR REPLACE FUNCTION trg_fecha_comentario()
  RETURNS TRIGGER
  LANGUAGE plpgsql
  AS $$
  BEGIN
    NEW.fecha_comentario := CURRENT_DATE;
    RETURN NEW;
  END;
  $$;

CREATE TRIGGER trg_fecha_comentario
  BEFORE INSERT ON comentario
  FOR EACH ROW
  EXECUTE FUNCTION trg_fecha_comentario();

```

5. Evitar reservas de aula con horario superpuesto





















```
CREATE OR REPLACE FUNCTION trg_validar_reserva()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN
    IF EXISTS (
        SELECT 1 FROM reserva_aula
        WHERE id_curso = NEW.id_curso
        AND fecha_reserva = NEW.fecha_reserva
        AND (
            (NEW.hora_inicio_reserva BETWEEN hora_inicio_reserva AND
            hora_fin_reserva)
            OR (NEW.hora_fin_reserva BETWEEN hora_inicio_reserva AND
            hora_fin_reserva)
        )
    ) THEN
        RAISE EXCEPTION 'Conflicto de horario en la reserva';
    END IF;





















    RETURN NEW;
END;
$$;
```

```
CREATE TRIGGER trg_validar_reserva
BEFORE INSERT ON reserva_aula
FOR EACH ROW
EXECUTE FUNCTION trg_validar_reserva();
```

3. Base de datos con Datos

	id_usuario	nombre_usuario	apellido	correo	contraseña	fecha_nacimiento	foto
1		Juan	Perez	juan@email.com	123456	NULL	NULL
2		Carlos	Serzuri Calizaya	sarzuricarlos60@gmail.com	12345678	2004-07-20	https://bapolkminyclqpdfjex.supabase.o
3		Oliver Saul	Garcia Guzman	olicho123@gmail.com	123456789	2004-12-26	https://bapolkminyclqpdfjex.supabase.o
4		José Carlos	Salvatierra Colque	jossalva64@gmail.com	jossito103salvatierra	1994-10-28	NULL
5		Wendy	Carvajal Lima	wendycarvajal76@gmail.com	wendy987carva	1993-06-10	NULL
6		Juan	Pérez	juan.perez@email.com	pt32da	1990-06-16	NULL
7		María	Gómez	maria.gomez@email.com	pdewf1	1986-08-22	NULL
8		Carlos	López	carlos.lopez@email.com	pdgdr1	1992-03-10	NULL
9		Ana	Martínez	ana.martinez@email.com	433csed	1988-11-30	NULL
10		Pedro	Rodríguez	pedro.rodriguez@email.com	af43	1996-07-18	NULL
11		Laura	García	laura.garcia@email.com	fv0th5	1991-12-05	NULL
12		Miguel	Hernández	miguel.hernandez@email.com	4345623	1987-04-26	NULL
13		Sofía	Díaz	sofia.diaz@email.com	bfgh014	1993-09-12	NULL

	 id_curso int4	 id_foro int4	titulo_curso varchar	descripcion_curso text	temario text
<input type="checkbox"/>	3	1 	Python para Principiantes	Curso introductorio de Python	Variables, estructuras de control,
<input type="checkbox"/>	4	2 	SQL Master	Dominio completo de SQL	Consultas, joins, subconsultas
<input type="checkbox"/>	5	3 	Desarrollo Web Full Stack	HTML, CSS, JavaScript, Node.js	Frontend y backend development
<input type="checkbox"/>	6	4 	Machine Learning con Python	Introducción al ML	Algoritmos, pandas, scikit-learn
<input type="checkbox"/>	7	5 	Android Development	Desarrollo de apps Android	Kotlin, Android Studio, APIs
<input type="checkbox"/>	8	6 	DevOps Fundamentals	Introducción a DevOps	Docker, Kubernetes, CI/CD
<input type="checkbox"/>	9	7 	Cybersecurity Basics	Conceptos de seguridad	Criptografía, pentesting
<input type="checkbox"/>	10	8 	AWS Cloud Practitioner	Certificación AWS	EC2, S3, IAM, CloudFront
<input type="checkbox"/>	11	9 	Data Analysis with R	Análisis de datos con R	Estadística, visualización
<input type="checkbox"/>	12	10 	UI/UX Design Principles	Diseño de interfaces	Figma, prototipado, research
<input type="checkbox"/>	13	11 	Unity Game Development	Desarrollo de juegos con Unity	C#, game physics, assets
<input type="checkbox"/>	14	12 	Blockchain Development	Smart contracts y DApps	Solidity, Ethereum, Web3
<input type="checkbox"/>	15	13 	IoT Programming	Programación para IoT	Arduino, Raspberry Pi, sensors
<input type="checkbox"/>	16	14 	Big Data with Hadoop	Procesamiento Big Data	HDFS, MapReduce, Spark
<input type="checkbox"/>	17	15 	Advanced Python	Python avanzado	Decoradores, generators, async
<input type="checkbox"/>	18	16 	Java Enterprise Edition	Java EE development	Servlets, JSP, EJB
<input type="checkbox"/>	19	17 	React Native Mobile	Apps móviles con React Native	Components, navigation, APIs
<input type="checkbox"/>	20	18 	C# .NET Core	Desarrollo con .NET Core	ASP.NET, Entity Framework

	 id_evaluacion int4	 id_modulo int4	notas int4	fecha_subida date
<input type="checkbox"/>	1	21 	85	2024-01-30
<input type="checkbox"/>	2	22 	90	2024-02-05
<input type="checkbox"/>	3	23 	78	2024-02-10
<input type="checkbox"/>	4	24 	92	2024-02-15
<input type="checkbox"/>	5	25 	88	2024-02-20
<input type="checkbox"/>	6	26 	75	2024-02-25
<input type="checkbox"/>	7	27 	95	2024-03-01
<input type="checkbox"/>	8	28 	82	2024-03-05
<input type="checkbox"/>	9	29 	79	2024-03-10
<input type="checkbox"/>	10	30 	91	2024-03-15
<input type="checkbox"/>	11	31 	87	2024-03-20
<input type="checkbox"/>	12	32 	84	2024-03-25
<input type="checkbox"/>	13	33 	93	2024-04-01
<input type="checkbox"/>	14	34 	76	2024-04-05
<input type="checkbox"/>	15	35 	89	2024-04-10
<input type="checkbox"/>	16	36 	94	2024-04-15
<input type="checkbox"/>	17	37 	81	2024-04-20
<input type="checkbox"/>	18	38 	86	2024-04-25

4. Funciones para CRUD

ASIGNACION_USUARIO_ROL

CREATE (INSERT):

```
INSERT INTO ASIGNACION_USUARIO_ROL (ID_ASIGNACION, ID_USUARIO, ID_ROL)
VALUES (valor_id_asignacion, valor_id_usuario, valor_id_rol);
```

READ (SELECT):

```
SELECT * FROM ASIGNACION_USUARIO_ROL;
SELECT * FROM ASIGNACION_USUARIO_ROL WHERE ID_ASIGNACION = valor_id_asignacion;
```

UPDATE (UPDATE):

```
UPDATE ASIGNACION_USUARIO_ROL
SET ID_USUARIO = nuevo_id_usuario, ID_ROL = nuevo_id_rol
WHERE ID_ASIGNACION = valor_id_asignacion;
```

DELETE (DELETE):

```
DELETE FROM ASIGNACION_USUARIO_ROL
WHERE ID_ASIGNACION = valor_id_asignacion;
```

ASIGNA_AULA

CREATE (INSERT):

```
INSERT INTO ASIGNA_AULA (ID_ASIGNA_AULA, ID_HORARIO, ID_AULA)
VALUES (valor_id_asigna_aula, valor_id_horario, valor_id_aula);
```

READ (SELECT):

```
SELECT * FROM ASIGNA_AULA;
SELECT * FROM ASIGNA_AULA WHERE ID_ASIGNA_AULA = valor_id_asigna_aula;
```

UPDATE (UPDATE):

```
UPDATE ASIGNA_AULA
SET ID_HORARIO = nuevo_id_horario, ID_AULA = nuevo_id_aula
WHERE ID_ASIGNA_AULA = valor_id_asigna_aula;
```

DELETE (DELETE):

```
DELETE FROM ASIGNA_AULA
WHERE ID_ASIGNA_AULA = valor_id_asigna_aula;
```

AULA

CREATE (INSERT):

```
INSERT INTO AULA (ID_AULA, NOMBRE_AULA, UBICACION)
VALUES (valor_id_aula, valor_nombre_aula, valor_ubicacion);
```

READ (SELECT):

```

SELECT * FROM AULA;
SELECT * FROM AULA WHERE ID_AULA = valor_id_aula;

UPDATE (UPDATE):

UPDATE AULA
SET NOMBRE_AULA = nuevo_nombre_aula, UBICACION = nuevo_ubicacion
WHERE ID_AULA = valor_id_aula;

DELETE (DELETE):

DELETE FROM AULA
WHERE ID_AULA = valor_id_aula;

CANJE
CREATE (INSERT):

INSERT INTO CANJE (ID_CANJE, ID_RECOMPENSA, ID_USUARIO, FECHA_CANJE)
VALUES (valor_id_canje, valor_id_recompensa, valor_id_usuario, valor_fecha_canje);

READ (SELECT):

SELECT * FROM CANJE;
SELECT * FROM CANJE WHERE ID_CANJE = valor_id_canje;

UPDATE (UPDATE):

UPDATE CANJE
SET ID_RECOMPENSA = nuevo_id_recompensa, ID_USUARIO = nuevo_id_usuario, FECHA_CANJE =
nuevo_fecha_canje
WHERE ID_CANJE = valor_id_canje;

DELETE (DELETE):

DELETE FROM CANJE
WHERE ID_CANJE = valor_id_canje;

CERTIFICADO
CREATE (INSERT):

INSERT INTO CERTIFICADO (ID_CERTIFICADO, ID_CURSO, ID_TIPO)
VALUES (valor_id_certificado, valor_id_curso, valor_id_tipo);

READ (SELECT):

SELECT * FROM CERTIFICADO;
SELECT * FROM CERTIFICADO WHERE ID_CERTIFICADO = valor_id_certificado;

UPDATE (UPDATE):

UPDATE CERTIFICADO
SET ID_CURSO = nuevo_id_curso, ID_TIPO = nuevo_id_tipo
WHERE ID_CERTIFICADO = valor_id_certificado;

DELETE (DELETE):

```



```
DELETE FROM CERTIFICADO
WHERE ID_CERTIFICADO = valor_id_certificado;
```

COMENTARIO

CREATE (INSERT):

```
INSERT INTO COMENTARIO (ID_COMENTARIO, ID_USUARIO, ID_FORO, CONTENIDO,
FECHA_COMENTARIO)
VALUES (valor_id_comentario, valor_id_usuario, valor_id_foro, valor_contenido,
valor_fecha_comentario);
```

READ (SELECT):

```
SELECT * FROM COMENTARIO;
SELECT * FROM COMENTARIO WHERE ID_COMENTARIO = valor_id_comentario;
```

UPDATE (UPDATE):

```
UPDATE COMENTARIO
SET ID_USUARIO = nuevo_id_usuario, ID_FORO = nuevo_id_foro, CONTENIDO = nuevo_contenido,
FECHA_COMENTARIO = nuevo_fecha_comentario
WHERE ID_COMENTARIO = valor_id_comentario;
```

DELETE (DELETE):

```
DELETE FROM COMENTARIO
WHERE ID_COMENTARIO = valor_id_comentario;
```

CURSO

CREATE (INSERT):

```
INSERT INTO CURSO (ID_CURSO, ID_FORO, TITULO_CURSO, DESCRIPCION_CURSO, TEMARIO,
DURACION, MODALIDAD, HORARIO_GENERAL, COSTO, CUPOS)
VALUES (valor_id_curso, valor_id_foro, valor_titulo_curso, valor_descripcion_curso, valor_temario,
valor_duracion, valor_modalidad, valor_horario_general, valor_costo, valor_cupos);
```

READ (SELECT):

```
SELECT * FROM CURSO;
SELECT * FROM CURSO WHERE ID_CURSO = valor_id_curso;
```

UPDATE (UPDATE):

```
UPDATE CURSO
SET ID_FORO = nuevo_id_foro, TITULO_CURSO = nuevo_titulo_curso, DESCRIPCION_CURSO =
nuevo_descripcion_curso, TEMARIO = nuevo_temario, DURACION = nuevo_duracion, MODALIDAD =
nuevo_modalidad, HORARIO_GENERAL = nuevo_horario_general, COSTO = nuevo_costo, CUPOS =
nuevo_cupos
WHERE ID_CURSO = valor_id_curso;
```

DELETE (DELETE):

```
DELETE FROM CURSO
WHERE ID_CURSO = valor_id_curso;
```

EVALUACION

CREATE (INSERT):

```
INSERT INTO EVALUACION (ID_EVALUACION, ID_MODULO, NOTAS, FECHA_SUBIDA)
VALUES (valor_id_evaluacion, valor_id_modulo, valor_notas, valor_fecha_subida);
```

READ (SELECT):

```
SELECT * FROM EVALUACION;
SELECT * FROM EVALUACION WHERE ID_EVALUACION = valor_id_evaluacion;
```

UPDATE (UPDATE):

```
UPDATE EVALUACION
SET ID_MODULO = nuevo_id_modulo, NOTAS = nuevo_notas, FECHA_SUBIDA = nuevo_fecha_subida
WHERE ID_EVALUACION = valor_id_evaluacion;
```

DELETE (DELETE):

```
DELETE FROM EVALUACION
WHERE ID_EVALUACION = valor_id_evaluacion;
```

EVENTO

CREATE (INSERT):

```
INSERT INTO EVENTO (ID_EVENTO, NOMBRE_EVENTO, FECHA_EVENTO, LUGAR,
DESCRIPCION_EVENTO)
VALUES (valor_id_evento, valor_nombre_evento, valor_fecha_evento, valor_lugar,
valor_descripcion_evento);
```

READ (SELECT):

```
SELECT * FROM EVENTO;
SELECT * FROM EVENTO WHERE ID_EVENTO = valor_id_evento;
```

UPDATE (UPDATE):

```
UPDATE EVENTO
SET NOMBRE_EVENTO = nuevo_nombre_evento, FECHA_EVENTO = nuevo_fecha_evento, LUGAR =
nuevo_lugar, DESCRIPCION_EVENTO = nuevo_descripcion_evento
WHERE ID_EVENTO = valor_id_evento;
```

DELETE (DELETE):

```
DELETE FROM EVENTO
WHERE ID_EVENTO = valor_id_evento;
```

EVENTO_USUARIO

CREATE (INSERT):

```
INSERT INTO EVENTO_USUARIO (ID_EVENTO_USUARIO, ID_EVENTO, ID_USUARIO)
VALUES (valor_id_evento_usuario, valor_id_evento, valor_id_usuario);
```

READ (SELECT):

```
SELECT * FROM EVENTO_USUARIO;  
SELECT * FROM EVENTO_USUARIO WHERE ID_EVENTO_USUARIO = valor_id_evento_usuario;
```

UPDATE (UPDATE):

```
UPDATE EVENTO_USUARIO  
SET ID_EVENTO = nuevo_id_evento, ID_USUARIO = nuevo_id_usuario  
WHERE ID_EVENTO_USUARIO = valor_id_evento_usuario;
```

DELETE (DELETE):

```
DELETE FROM EVENTO_USUARIO  
WHERE ID_EVENTO_USUARIO = valor_id_evento_usuario;
```

FORO

CREATE (INSERT):

```
INSERT INTO FORO (ID_FORO, TITULO_FORO, FECHA_CREACION, DESCRIPCION_FORO)  
VALUES (valor_id_foro, valor_titulo_foro, valor_fecha_creacion, valor_descripcion_foro);
```

READ (SELECT):

```
SELECT * FROM FORO;  
SELECT * FROM FORO WHERE ID_FORO = valor_id_foro;
```

UPDATE (UPDATE):

```
UPDATE FORO  
SET TITULO_FORO = nuevo_titulo_foro, FECHA_CREACION = nuevo_fecha_creacion,  
DESCRIPCION_FORO = nuevo_descripcion_foro  
WHERE ID_FORO = valor_id_foro;
```

DELETE (DELETE):

```
DELETE FROM FORO  
WHERE ID_FORO = valor_id_foro;
```

HORARIO

CREATE (INSERT):

```
INSERT INTO HORARIO (ID_HORARIO, ID_CURSO, DIA_SEMANA, HORA_INICIO_CLASE,  
HORA_FIN_CLASE, MODALIDADES)  
VALUES (valor_id_horario, valor_id_curso, valor_dia_semana, valor_hora_inicio_clase,  
valor_hora_fin_clase, valor_modalidades);
```

READ (SELECT):

```
SELECT * FROM HORARIO;  
SELECT * FROM HORARIO WHERE ID_HORARIO = valor_id_horario;
```

UPDATE (UPDATE):

```
UPDATE HORARIO  
SET ID_CURSO = nuevo_id_curso, DIA_SEMANA = nuevo_dia_semana, HORA_INICIO_CLASE =  
nuevo_hora_inicio_clase, HORA_FIN_CLASE = nuevo_hora_fin_clase, MODALIDADES =
```

```

nuevo_modalidades
WHERE ID_HORARIO = valor_id_horario;

DELETE (DELETE):

DELETE FROM HORARIO
WHERE ID_HORARIO = valor_id_horario;

INSCRIPCION
CREATE (INSERT):

INSERT INTO INSCRIPCION (ID_INSCRIPCION, ID_CURSO, ID_USUARIO, FECHA_INSCRIPCION,
ESTADO_INSCRIPCION)
VALUES (valor_id_inscripcion, valor_id_curso, valor_id_usuario, valor_fecha_inscripcion,
valor_estado_inscripcion);

READ (SELECT):

SELECT * FROM INSCRIPCION;
SELECT * FROM INSCRIPCION WHERE ID_INSCRIPCION = valor_id_inscripcion;

UPDATE (UPDATE):

UPDATE INSCRIPCION
SET ID_CURSO = nuevo_id_curso, ID_USUARIO = nuevo_id_usuario, FECHA_INSCRIPCION =
nuevo_fecha_inscripcion, ESTADO_INSCRIPCION = nuevo_estado_inscripcion
WHERE ID_INSCRIPCION = valor_id_inscripcion;

DELETE (DELETE):

DELETE FROM INSCRIPCION
WHERE ID_INSCRIPCION = valor_id_inscripcion;

INSIGNIA
CREATE (INSERT):

INSERT INTO INSIGNIA (ID_INSIGNIA, NOMBRE_INSIGNIA, DESCRIPCION_INSIGNIA,
REQUISITO_PARAMETRO)
VALUES (valor_id_insignia, valor_nombre_insignia, valor_descripcion_insignia,
valor_requisito_parametro);

READ (SELECT):

SELECT * FROM INSIGNIA;
SELECT * FROM INSIGNIA WHERE ID_INSIGNIA = valor_id_insignia;

UPDATE (UPDATE):

UPDATE INSIGNIA
SET NOMBRE_INSIGNIA = nuevo_nombre_insignia, DESCRIPCION_INSIGNIA =
nuevo_descripcion_insignia, REQUISITO_PARAMETRO = nuevo_requisito_parametro
WHERE ID_INSIGNIA = valor_id_insignia;

DELETE (DELETE):

```

```

DELETE FROM INSIGNIA
WHERE ID_INSIGNIA = valor_id_insignia;

LOGRO
CREATE (INSERT):

INSERT INTO LOGRO (ID_LOGRO, ID_USUARIO, ID_INSIGNIA)
VALUES (valor_id_logro, valor_id_usuario, valor_id_insignia);

READ (SELECT):

SELECT * FROM LOGRO;
SELECT * FROM LOGRO WHERE ID_LOGRO = valor_id_logro;

UPDATE (UPDATE):

UPDATE LOGRO
SET ID_USUARIO = nuevo_id_usuario, ID_INSIGNIA = nuevo_id_insignia
WHERE ID_LOGRO = valor_id_logro;

DELETE (DELETE):

DELETE FROM LOGRO
WHERE ID_LOGRO = valor_id_logro;

MODULOS
CREATE (INSERT):

INSERT INTO MODULOS (ID_MODULO, ID_CURSO, MATERIAS)
VALUES (valor_id_modulo, valor_id_curso, valor_materias);

READ (SELECT):

SELECT * FROM MODULOS;
SELECT * FROM MODULOS WHERE ID_MODULO = valor_id_modulo;

UPDATE (UPDATE):

UPDATE MODULOS
SET ID_CURSO = nuevo_id_curso, MATERIAS = nuevo_materias
WHERE ID_MODULO = valor_id_modulo;

DELETE (DELETE):

DELETE FROM MODULOS
WHERE ID_MODULO = valor_id_modulo;

PAGO
CREATE (INSERT):

INSERT INTO PAGO (ID_PAGO, ID_CURSO, ID_USUARIO, FECHA_PAGO, MONTO, METODO_PAGO)
VALUES (valor_id_pago, valor_id_curso, valor_id_usuario, valor_fecha_pago, valor_monto,
valor_metodo_pago);

READ (SELECT):

```

```

SELECT * FROM PAGO;
SELECT * FROM PAGO WHERE ID_PAGO = valor_id_pago;

UPDATE (UPDATE):

UPDATE PAGO
SET ID_CURSO = nuevo_id_curso, ID_USUARIO = nuevo_id_usuario, FECHA_PAGO =
nuevo_fecha_pago, MONTO = nuevo_monto, METODO_PAGO = nuevo_metodo_pago
WHERE ID_PAGO = valor_id_pago;

DELETE (DELETE):

DELETE FROM PAGO
WHERE ID_PAGO = valor_id_pago;

PERMISO
CREATE (INSERT):

INSERT INTO PERMISO (ID_PERMISO, NOMBRE_PERMISO, DESCRIPCION_PERMISO)
VALUES (valor_id_permiso, valor_nombre_permiso, valor_descripcion_permiso);

READ (SELECT):

SELECT * FROM PERMISO;
SELECT * FROM PERMISO WHERE ID_PERMISO = valor_id_permiso;

UPDATE (UPDATE):

UPDATE PERMISO
SET NOMBRE_PERMISO = nuevo_nombre_permiso, DESCRIPCION_PERMISO =
nuevo_descripcion_permiso
WHERE ID_PERMISO = valor_id_permiso;

DELETE (DELETE):

DELETE FROM PERMISO
WHERE ID_PERMISO = valor_id_permiso;

PROGRESO
CREATE (INSERT):

INSERT INTO PROGRESO (ID_PROGRESO, ID_EVALUACION, PORCENTAJE_PROGRESO,
ESTADO_PROGRESO)
VALUES (valor_id_progreso, valor_id_evaluacion, valor_porcentaje_progreso, valor_estado_progreso);

READ (SELECT):

SELECT * FROM PROGRESO;
SELECT * FROM PROGRESO WHERE ID_PROGRESO = valor_id_progreso;

UPDATE (UPDATE):

UPDATE PROGRESO
SET ID_EVALUACION = nuevo_id_evaluacion, PORCENTAJE_PROGRESO = nuevo_porcentaje_progreso,

```

```
ESTADO_PROGRESO = nuevo_estado_progreso
WHERE ID_PROGRESO = valor_id_progreso;
```

DELETE (DELETE):

```
DELETE FROM PROGRESO
WHERE ID_PROGRESO = valor_id_progreso;
```

PROGRESO_MODULO
CREATE (INSERT):

```
INSERT INTO PROGRESO_MODULO (ID_PROGRESO_MOD_, ID_MODULO,
PORCENTAJE_PROGR_MODULO, ESTADO_MODULO)
VALUES (valor_id_progreso_mod_, valor_id_modulo, valor_porcentaje_progr_modulo,
valor_estado_modulo);
```

READ (SELECT):

```
SELECT * FROM PROGRESO_MODULO;
SELECT * FROM PROGRESO_MODULO WHERE ID_PROGRESO_MOD_ = valor_id_progreso_mod_;
```

UPDATE (UPDATE):

```
UPDATE PROGRESO_MODULO
SET ID_MODULO = nuevo_id_modulo, PORCENTAJE_PROGR_MODULO =
nuevo_porcentaje_progr_modulo, ESTADO_MODULO = nuevo_estado_modulo
WHERE ID_PROGRESO_MOD_ = valor_id_progreso_mod_;
```

DELETE (DELETE):

```
DELETE FROM PROGRESO_MODULO
WHERE ID_PROGRESO_MOD_ = valor_id_progreso_mod_;
```

PUNTOS

CREATE (INSERT):

```
INSERT INTO PUNTOS (ID_PUNTOS, ID_USUARIO, ID_RECOMPENSA, SALDO_PUNTOS, MOTIVO,
PUNTOS_USADOS)
VALUES (valor_id_puntos, valor_id_usuario, valor_id_recompensa, valor_saldo_puntos, valor_motivo,
valor_puntos_usados);
```

READ (SELECT):

```
SELECT * FROM PUNTOS;
SELECT * FROM PUNTOS WHERE ID_PUNTOS = valor_id_puntos;
```

UPDATE (UPDATE):

```
UPDATE PUNTOS
SET ID_USUARIO = nuevo_id_usuario, ID_RECOMPENSA = nuevo_id_recompensa, SALDO_PUNTOS =
nuevo_saldo_puntos, MOTIVO = nuevo_motivo, PUNTOS_USADOS = nuevo_puntos_usados
WHERE ID_PUNTOS = valor_id_puntos;
```

DELETE (DELETE):

```

DELETE FROM PUNTOS
WHERE ID_PUNTOS = valor_id_puntos;

RECOMPENSA
CREATE (INSERT):

INSERT INTO RECOMPENSA (ID_RECOMPENSA, NOMBRE_RECOMPENSA, COSTO_PUNTOS)
VALUES (valor_id_recompensa, valor_nombre_recompensa, valor_costo_puntos);

READ (SELECT):

SELECT * FROM RECOMPENSA;
SELECT * FROM RECOMPENSA WHERE ID_RECOMPENSA = valor_id_recompensa;

UPDATE (UPDATE):

UPDATE RECOMPENSA
SET NOMBRE_RECOMPENSA = nuevo_nombre_recompensa, COSTO_PUNTOS = nuevo_costo_puntos
WHERE ID_RECOMPENSA = valor_id_recompensa;

DELETE (DELETE):

DELETE FROM RECOMPENSA
WHERE ID_RECOMPENSA = valor_id_recompensa;

RESERVA_AULA
CREATE (INSERT):

INSERT INTO RESERVA_AULA (ID_RESERVA, ID_CURSO, HORA_INICIO_RESERVA, HORA_FIN_RESERVA,
FECHA_RESERVA, ESTADO_RESERVA)
VALUES (valor_id_reserva, valor_id_curso, valor_hora_inicio_reserva, valor_hora_fin_reserva,
valor_fecha_reserva, valor_estado_reserva);

READ (SELECT):

SELECT * FROM RESERVA_AULA;
SELECT * FROM RESERVA_AULA WHERE ID_RESERVA = valor_id_reserva;

UPDATE (UPDATE):

UPDATE RESERVA_AULA
SET ID_CURSO = nuevo_id_curso, HORA_INICIO_RESERVA = nuevo_hora_inicio_reserva,
HORA_FIN_RESERVA = nuevo_hora_fin_reserva, FECHA_RESERVA = nuevo_fecha_reserva,
ESTADO_RESERVA = nuevo_estado_reserva
WHERE ID_RESERVA = valor_id_reserva;

DELETE (DELETE):

DELETE FROM RESERVA_AULA
WHERE ID_RESERVA = valor_id_reserva;

ROL
CREATE (INSERT):

```



```
INSERT INTO ROL (ID_ROL)
VALUES (valor_id_rol);
```

READ (SELECT):

```
SELECT * FROM ROL;
SELECT * FROM ROL WHERE ID_ROL = valor_id_rol;
```

UPDATE (UPDATE):

```
UPDATE ROL
SET ID_ROL = nuevo_id_rol
WHERE ID_ROL = valor_id_rol;
```

DELETE (DELETE):

```
DELETE FROM ROL
WHERE ID_ROL = valor_id_rol;
```

ROL_PERMISO

CREATE (INSERT):

```
INSERT INTO ROL_PERMISO (ID_ROL_PERMISO, ID_PERMISO, ID_ROL)
VALUES (valor_id_rol_permiso, valor_id_permiso, valor_id_rol);
```

READ (SELECT):

```
SELECT * FROM ROL_PERMISO;
SELECT * FROM ROL_PERMISO WHERE ID_ROL_PERMISO = valor_id_rol_permiso;
```

UPDATE (UPDATE):

```
UPDATE ROL_PERMISO
SET ID_PERMISO = nuevo_id_permiso, ID_ROL = nuevo_id_rol
WHERE ID_ROL_PERMISO = valor_id_rol_permiso;
```

DELETE (DELETE):

```
DELETE FROM ROL_PERMISO
WHERE ID_ROL_PERMISO = valor_id_rol_permiso;
```

TAREAS

CREATE (INSERT):

```
INSERT INTO TAREAS (ID_TAREA, ID_TEMAS, NOMBRE_TAREA, TIPO_TAREA)
VALUES (valor_id_tarea, valor_id_temas, valor_nombre_tarea, valor_tipo_tarea);
```

READ (SELECT):

```
SELECT * FROM TAREAS;
SELECT * FROM TAREAS WHERE ID_TAREA = valor_id_tarea;
```

UPDATE (UPDATE):

```

UPDATE TAREAS
SET ID_TEMAS = nuevo_id_temas, NOMBRE_TAREA = nuevo_nombre_tarea, TIPO_TAREA =
nuevo_tipo_tarea
WHERE ID_TAREA = valor_id_tarea;

DELETE (DELETE):

DELETE FROM TAREAS
WHERE ID_TAREA = valor_id_tarea;

TEMAS
CREATE (INSERT):

INSERT INTO TEMAS (ID_TEMAS, ID_MODULO, NOMBRE_TEMA, ESTADO_TEMA)
VALUES (valor_id_temas, valor_id_modulo, valor_nombre_tema, valor_estado_tema);

READ (SELECT):

SELECT * FROM TEMAS;
SELECT * FROM TEMAS WHERE ID_TEMAS = valor_id_temas;

UPDATE (UPDATE):

UPDATE TEMAS
SET ID_MODULO = nuevo_id_modulo, NOMBRE_TEMA = nuevo_nombre_tema, ESTADO_TEMA =
nuevo_estado_tema
WHERE ID_TEMAS = valor_id_temas;

DELETE (DELETE):

DELETE FROM TEMAS
WHERE ID_TEMAS = valor_id_temas;

TIPO_CURSO
CREATE (INSERT):

INSERT INTO TIPO_CURSO (ID_TIPO, ID_CURSO, NOMBRE_TIPO, DESCRIPCION_TIPO_CURSO)
VALUES (valor_id_tipo, valor_id_curso, valor_nombre_tipo, valor_descripcion_tipo_curso);

READ (SELECT):

SELECT * FROM TIPO_CURSO;
SELECT * FROM TIPO_CURSO WHERE ID_TIPO = valor_id_tipo;

UPDATE (UPDATE):

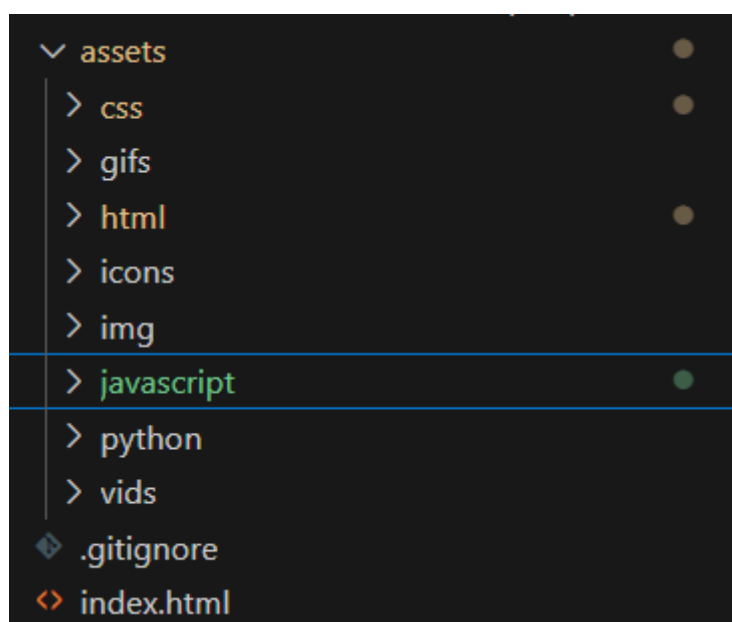
UPDATE TIPO_CURSO
SET ID_CURSO = nuevo_id_curso, NOMBRE_TIPO = nuevo_nombre_tipo, DESCRIPCION_TIPO_CURSO
= nuevo_descripcion_tipo_curso
WHERE ID_TIPO = valor_id_tipo;

DELETE (DELETE):

DELETE FROM TIPO_CURSO
WHERE ID_TIPO = valor_id_tipo;

```


6. Estructura del proyecto



7. Cronograma

Fechas	Descripción	Presentadores
10/09/25	Primer Avance del Hito 1	Todos
17/09/25	Diagrama ER	Vallejos Hidalgo Rody Brayan
24/09/25	Modelo ER,Requerimientos funcionales, Tablas del identidades y índices	Todos
01/10/25	Presentación del Hito 1	Todos
08/10/25	Modelo relacional	Garcia Guzman Oliver Saul
15/10/25	Refinamiento	Temprano Villarpando Dylan Alex
22/10/25	Procedimientos almacenados	Vallejos Hidalgo Rody Brayan
29/10/25	Desarrollo de Interfaz	Sarzuri Calizaya Carlos Andres Garcia Guzman Oliver Saul
05/11/25	Presentación del Hito 2	Todos
12/11/25	Base de Datos	Sarzuri Calizaya Carlos Andrés
19/11/25	Interfaz de Usuario	Garcia Guzman Oliver Saul
26/11/25	Pruebas de Funcionalidad	Temprano Villarpando Dylan Alex Vallejos Hidalgo Rody Brayan
03/12/25	Presentación del Hito 3	Todos
10/12/25	Informe Final Detallado	Todos
17/12/25	Presentación Final	Todos