



Proyecto de Hosting Educativo

Solicitante: I.T.S – Instituto Tecnológico Superior Arias Belparda.

Nombre Fantasía, de la nueva empresa: Blitz.

Grupo: TD4.

Turno: Matutino

Unidad Curricular: Téc. Redes y Software.

Nombres de los integrantes del grupo: Franco Behn, Patricio Montaño, Mariano Sosa.

Fecha de entrega: 14/11/2025

Instituto Tecnológico Superior Arias Belparda.
Blvr. José Batlle y Ordóñez 3570 esq. Gral. Flores – Montevideo.

Contenido

Elección y justificación del motor de base de datos	4
Comparativa entre MySQL y SQL Server	4
Ventajas de MySQL.....	4
Desventajas de SQL Server	4
Pasaje a Tablas.....	5
Usuario	5
Perfil	5
Contenido	6
Suscripcion	6
Pago	6
Mensaje	7
Diccionario de Datos	7
Normalización de la base de datos Educativa hasta la Tercera Forma Normal (3FN).....	9
Parte 1: Tablas Resultantes en 3FN	9
Parte 2: Estructura de cada Entidad	10
1. Usuario	10
2. Perfil.....	10
3. Contenido.....	10
4. Suscripcion	11
5. Pago	11
6. Mensaje	11
Notas Importantes.....	12
Cambios realizados respecto al documento original y según retroalimentación:.....	13
Script de MySQL	14

Auditoría y Registro de Logs en la Base de Datos	24
Lógica de nuestros logs y auditoría.....	24
¿Cómo gestionamos y resguardamos nuestros logs?.....	25
¿Cómo recuperamos nuestros logs?	25
Política de respaldos de la base de datos	26
Implementación del entorno con Docker	26
1. Ubicación y frecuencia	27
2. Validación y restauración	27
3. Tipos de respaldo recomendados y ejemplos	27
4. Buenas prácticas y políticas internas	28
Hoja testigo:	29
Firma del Profesor.....	29

Elección y justificación del motor de base de datos

Comparativa entre MySQL y SQL Server

Ventajas de MySQL

- Código abierto y gratuito:** Licencia gratuita para la mayoría de usos, reduciendo significativamente los costos de implementación.
- Simplicidad:** Instalación, configuración y administración más sencillas, permitiendo un despliegue rápido.
- Ligereza:** Menor consumo de recursos de hardware, ideal para optimizar costos de infraestructura.
- Multiplataforma:** Funciona en diversos sistemas operativos (Windows, Linux, macOS) sin problemas de compatibilidad.
- Comunidad activa:** Extensa comunidad de desarrolladores y abundante documentación gratuita.

Desventajas de SQL Server

- Costo elevado:** Requiere licencias comerciales costosas, especialmente para implementaciones a escala.
- Complejidad:** Curva de aprendizaje más pronunciada y configuración más compleja que MySQL.
- Requisitos de hardware:** Demanda mayores recursos computacionales para un rendimiento óptimo.
- Limitación de plataforma:** Principalmente orientado a entornos Windows, con limitaciones en otras plataformas.
- Sobrecarga de características:** Incluye funcionalidades avanzadas que resultan innecesarias para nuestro proyecto educativo.

- **Rendimiento en lecturas:** Excelente desempeño en operaciones de consulta, ideales para plataformas educativas.
- **Administración más compleja:** Requiere personal especializado para su mantenimiento y optimización.

Pasaje a Tablas

Usuario

- IdUsuario (PK)
- NombreUsuario
- Correo
- HashContrasena
- FechaRegistro
- EstadoCuenta
- TipoUsuario
- PreferenciasPrivacidad

Perfil

- IdPerfil (PK y FK de Usuario, 1:1)
- FotoPerfil
- Biografia
- EnlacesSociales
- EnlacePersonal

Contenido

- IdContenido (PK)
- IdAutor (FK, 1:N con Usuario)
- Titulo
- Descripcion
- TipoContenido
- UrlContenido
- FechaCreacion
- EstadoContenido

Suscripcion

- IdSuscripcion (PK)
- IdSuscriptor (FK, 1:N con Usuario)
- IdCreador (FK, 1:N con Usuario)
- Monto
- FechaPago
- EstadoPago

Pago

- IdPago (PK)
- IdPagador (FK, 1:N con Usuario)
- IdReceptor (FK, 1:N con Usuario)
- Monto
- FechaPago
- EstadoPago
- MetodoPago
- IdEmisor

Mensaje

- IdMensaje (PK)
- IdEmisor (FK, 1:N con Usuario)
- IdReceptor (FK, 1:N con Usuario)
- ContenidoMensaje
- FechaMensaje
- EstadoMensaje

Diccionario de Datos

Tabla	Campo	Tipo	Descripción
Usuario	IdUsuario	int (PK)	Identificador único del usuario
	NombreUsuario	varchar	Nombre público del usuario
	Correo	varchar	Email del usuario
	HashContrasena	varchar	Contraseña cifrada
	FechaRegistro	datetime	Fecha de registro
	EstadoCuenta	varchar	Estado (activo, suspendido, eliminado)
	TipoUsuario	varchar	Tipo de usuario (creador, suscriptor, admin)
	PreferenciasPrivacidad	varchar	Opciones de privacidad

Perfil	IdPerfil	int (PK, FK)	Igual a IdUsuario
	FotoPerfil	varchar	URL o ruta de la foto de perfil
	Biografia	varchar	Biografía o descripción
	EnlacesSociales	varchar	Enlaces a redes sociales (puede descomponerse)
	EnlacePersonal	varchar	Enlace personal del usuario
Contenido	IdContenido	int (PK)	Identificador del contenido
	IdAutor	int (FK)	Usuario creador del contenido
	Titulo	varchar	Título del contenido
	Descripcion	varchar	Descripción
	TipoContenido	varchar	Tipo
	UrlContenido	varchar	URL/almacenamiento seguro
	FechaCreacion	datetime	Fecha de creación
	EstadoContenido	varchar	Estado (público, privado, suscriptores)
Suscripcion	IdSuscripcion	int (PK)	Identificador de suscripción
	IdSuscriptor	int (FK)	Usuario que realiza la suscripción
	IdCreador	int (FK)	Usuario que recibe la suscripción
	Monto	decimal	Monto
	FechaPago	datetime	Fecha de pago
	EstadoPago	varchar	Estado
Pago	IdPago	int (PK)	Identificador de pago
	IdPagador	int (FK)	Usuario que paga
	IdReceptor	int (FK)	Usuario a quien se paga
	Monto	decimal	Monto de la transacción
	FechaPago	datetime	Fecha de pago
	EstadoPago	varchar	Estado del pago
	MetodoPago	varchar	Método de pago
	IdEmisor	int	Emisor

Mensaje	IdMensaje	int (PK)	Identificador de mensaje
	IdEmisor	int (FK)	Usuario que envía el mensaje
	IdReceptor	int (FK)	Usuario que recibe el mensaje
	ContenidoMensaje	varchar	Contenido del mensaje
	FechaMensaje	datetime	Fecha de envío del mensaje
	EstadoMensaje	varchar	Estado (leído, no leído)

Normalización de la base de datos Educativa hasta la Tercera Forma Normal (3FN)

Parte 1: Tablas Resultantes en 3FN

Nº	Tabla	Descripción
1	Usuario	Contiene la información básica y única de cada usuario.
2	Perfil	Información adicional y pública exclusiva de cada usuario (1:1 con Usuario).
3	Contenido	Recursos educativos generados por creadores (varios por usuario).
4	Suscripcion	Registro de suscripciones entre usuarios (dos referencias a Usuario).
5	Pago	Transacciones financieras (cada pago es pagador y receptor, ambos Usuario).
6	Mensaje	Sistema de mensajería privada entre usuarios (emisor y receptor).

Parte 2: Estructura de cada Entidad

1. Usuario

Atributo	Tipo / Clave	Descripción
IdUsuario	INT (PK)	Identificador único del usuario
NombreUsuario	VARCHAR	Nombre de usuario visible
Correo	VARCHAR	Correo electrónico registrado
HashContraseña	VARCHAR	Contraseña en hash
FechaRegistro	DATETIME	Fecha de creación en el sistema
EstadoCuenta	VARCHAR	Estado: activo, suspendido, eliminado
TipoUsuario	VARCHAR	Rol: creador, suscriptor, admin
PreferenciasPrivacidad	VARCHAR	Preferencias sobre privacidad

2. Perfil

Atributo	Tipo / Clave	Descripción
IdPerfil	INT (PK, FK)	Identificador de perfil (idéntico al usuario)
FotoPerfil	VARCHAR	Foto o avatar de usuario
Biografia	VARCHAR	Breve descripción personal o profesional
EnlacesSociales	VARCHAR	Enlaces a redes sociales
EnlacePersonal	VARCHAR	Página personal o profesional

3. Contenido

Atributo	Tipo / Clave	Descripción
IdContenido	INT (PK)	Identificador único del contenido
IdAutor	INT (FK)	Usuario creador del contenido
Titulo	VARCHAR	Título del recurso
Descripcion	VARCHAR	Breve descripción
TipoContenido	VARCHAR	Tipo: video, pdf, etc.
UrlContenido	VARCHAR	URL/ubicación segura
FechaCreacion	DATETIME	Fecha de creación
EstadoContenido	VARCHAR	Público, privado, sólo suscriptores

4. Suscripcion

Atributo	Tipo / Clave	Descripción
IdSuscripcion	INT (PK)	Identificador de la suscripción
IdSuscriptor	INT (FK)	Usuario que realiza la suscripción
IdCreador	INT (FK)	Usuario que la recibe (rol de creador)
Monto	DECIMAL	Monto del pago de la suscripción
FechaPago	DATETIME	Fecha de pago asociado a la suscripción
EstadoPago	VARCHAR	Estado: pagado, pendiente, cancelado, etc.

5. Pago

Atributo	Tipo / Clave	Descripción
IdPago	INT (PK)	Identificador del pago
IdPagador	INT (FK)	Usuario que efectúa el pago
IdReceptor	INT (FK)	Usuario que recibe el pago
Monto	DECIMAL	Monto de la transacción
FechaPago	DATETIME	Fecha en que se realiza
EstadoPago	VARCHAR	Estado de la transacción
MetodoPago	VARCHAR	Canal: tarjeta, PayPal, etc.
IdEmisor	INT	Emisor del pago (si aplica)

6. Mensaje

Atributo	Tipo / Clave	Descripción
IdMensaje	INT (PK)	Identificador único del mensaje
IdEmisor	INT (FK)	Usuario que envía el mensaje
IdReceptor	INT (FK)	Usuario que lo recibe
ContenidoMensaje	VARCHAR	Texto del mensaje
FechaMensaje	DATETIME	Fecha/hora de envío
EstadoMensaje	VARCHAR	Estado: leído, no leído, etc.

Notas Importantes

- Todas las entidades de la base están **normalizadas hasta 3FN**:
 - No existen atributos multivalorados, grupos repetitivos ni dependencias transitivas.
 - Cada atributo depende completamente de la clave primaria de su entidad.
 - Si se requiere mayor granularidad, el campo EnlacesSociales puede descomponerse en una nueva tabla EnlaceSocial (cumpliendo 1FN y 3FN estrictas).
- Las claves foráneas representan relaciones de *totalidad*, y se emplean solo en el modelo físico (tablas) para asegurar la integridad referencial.
- **No se utiliza la entidad Consentimiento** por indicación de corrección.
- El modelo facilita la privacidad y seguridad de los datos, permitiendo implementar controles específicos y auditoría.
- Los nombres de los atributos pueden adecuarse a las convenciones requeridas por el sistema.
- Se han hecho aclaraciones didácticas sobre la finalidad de las relaciones, la lógica de totalidades y la representación de doble relación (dos roles distintos) en entidades como Suscripcion, Pago y Mensaje.

Cambios realizados respecto al documento original y según retroalimentación:

1. D.E-R.:

- Uso correcto y explícito de relaciones 1:N y 1:1 con totalidad.
- Eliminación de claves foráneas del D.E-R.; solo se muestran en las tablas físicas.
- Doble relación en entidades donde corresponde (ej: Suscripcion, Pago, Mensaje con Usuario).
- DER ajustado con notación de Chen vertical clara, nombres sin tildes/acentos innecesarios.

2. Script y Diccionario de Datos:

- Tipos de dato NVARCHAR convertidos a VARCHAR para MySQL.
- Uso consistente de tipos y convenciones de nombres.

3. Eliminación de toda referencia a la tabla Consentimiento en tablas, diccionario y DER.

4. Docker:

- Incluida una guía docker-compose para facilitar pruebas y despliegue portátil.

5. Notas/Guías:

- Incorporadas explicaciones sobre la construcción del DER, el uso de totalidades y la omisión de FKs en el modelo conceptual.
- Explicada claramente la representación de dobles relaciones entre Usuario y entidades dependientes.

6. Normas de normalización reforzadas y adaptadas a la realidad del proyecto.

Script de MySQL

-- Creación de la base de datos y uso de la misma.

CREATE DATABASE IF NOT EXISTS Project;

USE Project;

-- =====

-- TABLAS PRINCIPALES

-- =====

-- Tabla de Usuarios.

-- Cada usuario contiene su identificador único, nombre de usuario que se utiliza como "login", correo electrónico y atributos de seguridad y rol.

CREATE TABLE Usuarios (

IdUsuario **INT PRIMARY KEY AUTO_INCREMENT**,
NombreUsuario **VARCHAR(16) UNIQUE NOT NULL**,
Correo **VARCHAR(64) NOT NULL**,
HashContraseña **VARCHAR(128) NOT NULL**,
FechaRegistro **DATETIME NOT NULL**,
EstadoCuenta **VARCHAR(20) NOT NULL**,
TipoUsuario **VARCHAR(20) NOT NULL**,
PreferenciasPrivacidad **VARCHAR(20) NOT NULL**

);

-- Tabla de Perfil (relación 1:1 con Usuarios; IdPerfil = IdUsuario).

-- La información del perfil puede contener foto, biografía y enlaces, es opcional respecto a la existencia de un Usuario.

CREATE TABLE Perfil (

IdPerfil **INT PRIMARY KEY**,
FotoPerfil **VARCHAR(256)**,
Biografia **VARCHAR(512)**,
EnlacesSociales **VARCHAR(256)**,
EnlacePersonal **VARCHAR(256)**,
FOREIGN KEY (IdPerfil) **REFERENCES** Usuarios(IdUsuario)

);

-- Tabla de Contenido (relación 1:N con Usuarios).
 -- Un usuario (creador) puede crear varios contenidos.

CREATE TABLE Contenido (

```

IdContenido INT PRIMARY KEY AUTO_INCREMENT,
IdAutor INT NOT NULL,
Titulo VARCHAR(64) NOT NULL,
Descripcion VARCHAR(2048),
TipoContenido VARCHAR(32),
UrlContenido VARCHAR(256) NOT NULL,
FechaCreacion DATETIME NOT NULL,
EstadoContenido VARCHAR(20) NOT NULL,
FOREIGN KEY (IdAutor) REFERENCES Usuarios(IdUsuario)
);
  
```

-- Tabla de Suscripcion (relación N:1, doble relación a Usuarios).
 -- Permite gestionar suscriptores y creadores, incluyendo la información del pago.

CREATE TABLE Suscripcion (

```

IdSuscripcion INT PRIMARY KEY AUTO_INCREMENT,
IdSuscriptor INT NOT NULL,
IdCreador INT NOT NULL,
Monto DECIMAL(10,2) NOT NULL,
FechaPago DATETIME NOT NULL,
EstadoPago VARCHAR(20) NOT NULL,
FOREIGN KEY (IdSuscriptor) REFERENCES Usuarios(IdUsuario),
FOREIGN KEY (IdCreador) REFERENCES Usuarios(IdUsuario)
);
  
```

-- Tabla de Pago (relación N:1, doble relación a Usuarios).
 -- Registra los pagos realizados entre usuarios, permitiendo diferentes métodos y emisor.

CREATE TABLE Pago (

```

IdPago INT PRIMARY KEY AUTO_INCREMENT,
IdPagador INT NOT NULL,
IdReceptor INT NOT NULL,
Monto DECIMAL(10,2) NOT NULL,
FechaPago DATETIME NOT NULL,
EstadoPago VARCHAR(20) NOT NULL,
MetodoPago VARCHAR(32) NOT NULL,
IdEmisor INT,
FOREIGN KEY (IdPagador) REFERENCES Usuarios(IdUsuario),
FOREIGN KEY (IdReceptor) REFERENCES Usuarios(IdUsuario)
);
  
```

-- Tabla de Mensaje (relación N:1, doble relación a Usuarios).
 -- Maneja el sistema de mensajería entre usuarios, permitiendo mensajes bidireccionales.

CREATE TABLE Mensaje (

```

IdMensaje INT PRIMARY KEY AUTO_INCREMENT,
IdEmisor INT NOT NULL,
IdReceptor INT NOT NULL,
ContenidoMensaje VARCHAR(1024) NOT NULL,
FechaMensaje DATETIME NOT NULL,
EstadoMensaje VARCHAR(20) NOT NULL,
FOREIGN KEY (IdEmisor) REFERENCES Usuarios(IdUsuario),
FOREIGN KEY (IdReceptor) REFERENCES Usuarios(IdUsuario)
);
```

=====
 -- VISTAS Y FUNCIONES PARA ROLES Y PERMISOS
 =====

-- Vista Roles: mapea el tipo de usuario a su rol funcional dentro del sistema.

CREATE OR REPLACE VIEW Roles **AS**

SELECT

```

IdUsuario,
NombreUsuario,
TipoUsuario,
CASE
  WHEN TipoUsuario = 'Creador' THEN 'creador de contenido'
  WHEN TipoUsuario = 'Suscriptor' THEN 'cliente'
  ELSE 'administrador'
END AS Rol
FROM Usuarios;
```

-- RolePermissions: define los permisos asociados a cada rol.

CREATE OR REPLACE VIEW RolePermissions **AS**

```
SELECT 'cliente' AS Rol,  
       1 AS can_view_content,  
       0 AS can_create_content,  
       0 AS can_manage_users,  
       0 AS can_manage_site,  
       1 AS can_pay,  
       1 AS can_message
```

UNION ALL

```
SELECT 'creador de contenido',  
       1,  
       1,  
       0,  
       0,  
       1,  
       1
```

UNION ALL

```
SELECT 'administrador',  
       1,  
       1,  
       1,  
       1,  
       1,  
       1;
```

-- UserPermissions: combina cada usuario con los permisos correspondientes a su rol.

CREATE OR REPLACE VIEW UserPermissions **AS**

```
SELECT R.IdUsuario,  
      R.NombreUsuario,  
      R.TipoUsuario,  
      RP.Rol,  
      RP.can_view_content,  
      RP.can_create_content,  
      RP.can_manage_users,  
      RP.can_manage_site,  
      RP.can_pay,  
      RP.can_message  
FROM Roles R  
LEFT JOIN RolePermissions RP ON R.Rol = RP.Rol;
```

-- Función HasPermission(userId, permisoClave): devuelve 1 si el usuario tiene el permiso solicitado para la clave, 0 en caso contrario.

-- Permisos válidos: 'view','create','manage_users','manage_site','pay','message'

DELIMITER //

CREATE FUNCTION HasPermission(pUserId **INT**, pPerm **VARCHAR(64)**)

RETURNS TINYINT(1)

DETERMINISTIC

READS SQL DATA

BEGIN

DECLARE v_result **TINYINT(1) DEFAULT 0;**

SELECT

CASE pPerm

WHEN 'view' **THEN** UP.can_view_content

WHEN 'create' **THEN** UP.can_create_content

WHEN 'manage_users' **THEN** UP.can_manage_users

WHEN 'manage_site' **THEN** UP.can_manage_site

WHEN 'pay' **THEN** UP.can_pay

WHEN 'message' **THEN** UP.can_message

ELSE 0

END

INTO v_result

FROM UserPermissions UP

WHERE UP.IdUsuario = pUserId

LIMIT 1;

RETURN IFNULL(v_result, 0);

END//

DELIMITER ;

```
-- =====
```

-- PROCEDIMIENTOS, TRIGGERS Y EVENTOS UTILES

```
-- =====
```

DELIMITER //

-- Procedimiento almacenado para dar de baja lógica a un usuario (no borrado físico).
-- Cambia el estado de la cuenta a 'Inactivo'. Puede auditarse en una tabla de auditorías si se desea registrar motivos.

CREATE PROCEDURE DeactivateUser(**IN** pUserId **INT**, **IN** pReason **VARCHAR(512)**)

BEGIN

UPDATE Usuarios

SET EstadoCuenta = 'Inactivo'

WHERE IdUsuario = pUserId;

 -- Registro de la baja puede implementarse en tabla de auditoría separada, si se requiere para histórico.

END//

-- Trigger preventivo para evitar el borrado físico de usuarios.

-- Se exige la baja lógica por procedimiento almacenado.

CREATE TRIGGER Usuarios_prevent_delete

BEFORE DELETE ON Usuarios

FOR EACH ROW

BEGIN

SIGNAL SQLSTATE '45000'

SET MESSAGE_TEXT = 'DELETE no permitido en Usuarios. Use CALL
DeactivateUser(id, motivo) para desactivar la cuenta.';

END//

-- Evento para caducar suscripciones pagadas hace más de un año.

-- Debe estar habilitado el scheduler de eventos (SET GLOBAL event_scheduler = ON).

CREATE EVENT IF NOT EXISTS ExpireOldSubscriptions

ON SCHEDULE EVERY 1 DAY

DO

UPDATE Suscripcion

SET EstadoPago = 'Expirado'

WHERE EstadoPago = 'Pagado'

AND FechaPago < **DATE_SUB(NOW(), INTERVAL 1 YEAR)**;

DELIMITER ;

-- =====
-- DATOS DE PRUEBA (INSERTS)
-- =====

-- Ejemplo de alta de usuarios, perfiles, contenidos, suscripciones, pagos y mensajes.

-- Usuarios base de prueba (20 registros).

```
INSERT INTO Usuarios (NombreUsuario, Correo, HashContraseña, FechaRegistro,  
EstadoCuenta, TipoUsuario, PreferenciasPrivacidad) VALUES  
('Alice', 'alice@email.com', SHA2('passA', 256), '2025-09-01', 'Activo', 'Creador', 'Privado'),  
('Bob', 'bob@email.com', SHA2('passB', 256), '2025-09-02', 'Activo', 'Suscriptor', 'Publico'),  
('Carla', 'carla@email.com', SHA2('passC', 256), '2025-09-03', 'Activo', 'Creador',  
'Privado'),  
('Daniel', 'daniel@email.com', SHA2('passD', 256), '2025-09-04', 'Suspendido', 'Suscriptor',  
'Privado'),  
('Eva', 'eva@email.com', SHA2('passE', 256), '2025-09-05', 'Activo', 'Creador', 'Publico'),  
('Frank', 'frank@email.com', SHA2('passF', 256), '2025-09-06', 'Activo', 'Suscriptor',  
'Privado'),  
('Gina', 'gina@email.com', SHA2('passG', 256), '2025-09-07', 'Activo', 'Creador', 'Privado'),  
('Hector', 'hector@email.com', SHA2('passH', 256), '2025-09-08', 'Activo', 'Suscriptor',  
'Publico'),  
('Irene', 'irene@email.com', SHA2('passI', 256), '2025-09-09', 'Activo', 'Creador', 'Privado'),  
('Juan', 'juan@email.com', SHA2('passJ', 256), '2025-09-10', 'Activo', 'Suscriptor',  
'Privado'),  
('Karla', 'karla@email.com', SHA2('passK', 256), '2025-09-11', 'Activo', 'Creador', 'Publico'),  
('Luis', 'luis@email.com', SHA2('passL', 256), '2025-09-12', 'Activo', 'Suscriptor', 'Privado'),  
('Marina', 'marina@email.com', SHA2('passM', 256), '2025-09-13', 'Activo', 'Creador',  
'Privado'),  
('Nicolas', 'nicolas@email.com', SHA2('passN', 256), '2025-09-14', 'Activo', 'Suscriptor',  
'Publico'),  
('Olga', 'olga@email.com', SHA2('passO', 256), '2025-09-15', 'Activo', 'Creador', 'Privado'),  
('Pablo', 'pablo@email.com', SHA2('passP', 256), '2025-09-16', 'Activo', 'Suscriptor',  
'Privado'),  
('Quique', 'quique@email.com', SHA2('passQ', 256), '2025-09-17', 'Activo', 'Creador',  
'Publico'),  
('Rosa', 'rosa@email.com', SHA2('passR', 256), '2025-09-18', 'Activo', 'Suscriptor',  
'Privado'),  
('Samuel', 'samuel@email.com', SHA2('passS', 256), '2025-09-19', 'Activo', 'Creador',  
'Privado'),  
('Tania', 'tania@email.com', SHA2('passT', 256), '2025-09-20', 'Activo', 'Suscriptor',  
'Publico');
```

-- Perfiles asociados.



INSERT INTO Perfil (IdPerfil, FotoPerfil, Biografia, EnlacesSociales, EnlacePersonal) **VALUES**

(1, 'alice.jpg', 'Fotógrafa profesional', 'https://instagram.com/alice', 'https://alice.com'),
(2, 'bob.jpg', 'Amante de la tecnología', 'https://twitter.com/bob', **NULL**),
(3, 'carla.jpg', 'Escritora y bloguera', 'https://facebook.com/carla', 'https://carla.com'),
(4, 'daniel.jpg', 'Fan de los videojuegos', **NULL**, **NULL**),
(5, 'eva.jpg', 'Creadora de contenido fitness', 'https://instagram.com/eva',
'https://eva.com'),
(6, 'frank.jpg', 'Músico aficionado', 'https://soundcloud.com/frank', **NULL**),
(7, 'gina.jpg', 'Chef y repostera', 'https://youtube.com/gina', 'https://gina.com'),
(8, 'hector.jpg', 'Estudiante de ingeniería', **NULL**, **NULL**),
(9, 'irene.jpg', 'Diseñadora gráfica', 'https://behance.net/irene', 'https://irene.com'),
(10, 'juan.jpg', 'Coleccionista de cómics', **NULL**, **NULL**),
(11, 'karla.jpg', 'Streamer de videojuegos', 'https://twitch.tv/karla', 'https://karla.com'),
(12, 'luis.jpg', 'Creador de podcasts', 'https://spotify.com/luis', **NULL**),
(13, 'marina.jpg', 'Artista digital', 'https://deviantart.com/marina', 'https://marina.com'),
(14, 'nicolas.jpg', 'Fotógrafo amateur', **NULL**, **NULL**),
(15, 'olga.jpg', 'Creadora de tutoriales', 'https://youtube.com/olga', 'https://olga.com'),
(16, 'pablo.jpg', 'Fan de la ciencia ficción', **NULL**, **NULL**),
(17, 'quique.jpg', 'Creador de memes', 'https://twitter.com/quique', 'https://quique.com'),
(18, 'rosa.jpg', 'Estudiante de medicina', **NULL**, **NULL**),
(19, 'samuel.jpg', 'Creador de contenido educativo', 'https://youtube.com/samuel',
'https://samuel.com'),
(20, 'tania.jpg', 'Amante de los animales', **NULL**, **NULL**);

-- Ejemplo de contenidos.

INSERT INTO Contenido (IdAutor, Titulo, Descripcion, TipoContenido, UrlContenido, FechaCreacion, EstadoContenido) **VALUES**

(1, 'Fotografía de paisaje', 'Paisaje natural al amanecer', 'Imagen',
'https://alice.com/paisaje', '2025-09-01', 'Publicado'),
(3, 'Artículo sobre escritura', 'Consejos para escritores', 'Artículo',
'https://carla.com/escritura', '2025-09-03', 'Publicado'),
(5, 'Rutina de ejercicios', 'Entrenamiento para principiantes', 'Video',
'https://eva.com/ejercicio', '2025-09-05', 'Publicado'),
(7, 'Receta de pastel', 'Cómo hacer pastel de chocolate', 'Video', 'https://gina.com/pastel',
'2025-09-07', 'Publicado'),
(9, 'Diseño de logo', 'Logo para empresa ficticia', 'Imagen', 'https://irene.com/logo',
'2025-09-09', 'Publicado'),
(11, 'Stream de juego', 'Jugando Minecraft en vivo', 'Video', 'https://karla.com/minecraft',
'2025-09-11', 'Publicado'),
(13, 'Ilustración digital', 'Arte conceptual', 'Imagen', 'https://marina.com/arte', '2025-09-13', 'Publicado'),
(15, 'Tutorial de Excel', 'Cómo usar fórmulas', 'Video', 'https://olga.com/excel', '2025-09-15', 'Publicado'),

(17, 'Meme viral', 'Meme sobre exámenes', 'Imagen', 'https://quique.com/meme', '2025-09-17', 'Publicado'),
 (19, 'Video educativo', 'Historia de México', 'Video', 'https://samuel.com/historia', '2025-09-19', 'Publicado'),
 (1, 'Retrato artístico', 'Retrato en blanco y negro', 'Imagen', 'https://alice.com/retrato', '2025-09-02', 'Publicado'),
 (5, 'Consejos de nutrición', 'Alimentación saludable', 'Artículo', 'https://eva.com/nutricion', '2025-09-06', 'Publicado'),
 (7, 'Receta de galletas', 'Galletas de avena', 'Video', 'https://gina.com/galletas', '2025-09-08', 'Publicado'),
 (11, 'Stream de Fortnite', 'Jugando Fortnite en vivo', 'Video', 'https://karla.com/fortnite', '2025-09-12', 'Publicado'),
 (13, 'Animación digital', 'Animación de personaje', 'Video', 'https://marina.com/animacion', '2025-09-14', 'Publicado');

-- Ejemplo de suscripciones.

```
INSERT INTO Suscripcion (IdSuscriptor, IdCreador, Monto, FechaPago, EstadoPago) VALUES
(2, 1, 50.00, '2025-09-02', 'Pagado'),
(4, 3, 30.00, '2025-09-04', 'Pagado'),
(6, 5, 40.00, '2025-09-06', 'Pagado'),
(8, 7, 25.00, '2025-09-08', 'Pagado'),
(10, 9, 35.00, '2025-09-10', 'Pagado'),
(12, 11, 45.00, '2025-09-12', 'Pagado'),
(14, 13, 55.00, '2025-09-14', 'Pagado'),
(16, 15, 60.00, '2025-09-16', 'Pagado'),
(18, 17, 20.00, '2025-09-18', 'Pagado'),
(20, 19, 70.00, '2025-09-20', 'Pagado');
```

-- Ejemplo de pagos.

```
INSERT INTO Pago (IdPagador, IdReceptor, Monto, FechaPago, EstadoPago, MetodoPago, IdEmisor) VALUES
(2, 1, 50.00, '2025-09-02', 'Completado', 'Tarjeta', 2),
(4, 3, 30.00, '2025-09-04', 'Completado', 'Paypal', 4),
(6, 5, 40.00, '2025-09-06', 'Completado', 'Tarjeta', 6),
(8, 7, 25.00, '2025-09-08', 'Completado', 'Paypal', 8),
(10, 9, 35.00, '2025-09-10', 'Completado', 'Tarjeta', 10),
(12, 11, 45.00, '2025-09-12', 'Completado', 'Paypal', 12),
(14, 13, 55.00, '2025-09-14', 'Completado', 'Tarjeta', 14),
(16, 15, 60.00, '2025-09-16', 'Completado', 'Paypal', 16),
(18, 17, 20.00, '2025-09-18', 'Completado', 'Tarjeta', 18),
(20, 19, 70.00, '2025-09-20', 'Completado', 'Paypal', 20);
```

-- Ejemplo de mensajes privados.

INSERT INTO Mensaje (IdEmisor, IdReceptor, ContenidoMensaje, FechaMensaje, EstadoMensaje) **VALUES**

```
(2, 1, 'Hola Alice, me gusta tu trabajo!', '2025-09-02 10:00', 'Leido'),
(4, 3, 'Carla, ¿cuándo publicas nuevo artículo?', '2025-09-04 11:00', 'Leido'),
(6, 5, 'Eva, ¿tienes rutinas para principiantes?', '2025-09-06 12:00', 'Leido'),
(8, 7, 'Gina, tu receta fue un éxito!', '2025-09-08 13:00', 'Leido'),
(10, 9, 'Irene, ¿aceptas encargos?', '2025-09-10 14:00', 'Leido'),
(12, 11, 'Karla, ¿harás stream hoy?', '2025-09-12 15:00', 'Leido'),
(14, 13, 'Marina, tu arte es genial!', '2025-09-14 16:00', 'Leido'),
(16, 15, 'Olga, ¿más tutoriales pronto?', '2025-09-16 17:00', 'Leido'),
(18, 17, 'Quique, tus memes son los mejores!', '2025-09-18 18:00', 'Leido'),
(20, 19, 'Samuel, ¿qué tema será el próximo video?', '2025-09-20 19:00', 'Leido'),
(1, 2, 'Gracias por tu apoyo, Bob!', '2025-09-02 20:00', 'Leido'),
(3, 4, '¡Pronto publicaré más!', '2025-09-04 21:00', 'Leido'),
(5, 6, 'Sí, tengo rutinas para todos.', '2025-09-06 22:00', 'Leido'),
(7, 8, 'Me alegra que te gustara!', '2025-09-08 23:00', 'Leido'),
(9, 10, 'Claro, mándame detalles.', '2025-09-10 09:00', 'Leido'),
(11, 12, 'Sí, a las 8pm!', '2025-09-12 08:00', 'Leido'),
(13, 14, '¡Gracias por tu comentario!', '2025-09-14 07:00', 'Leido'),
(15, 16, 'Sí, pronto más tutoriales.', '2025-09-16 06:00', 'Leido'),
(17, 18, '¡Gracias por seguirme!', '2025-09-18 05:00', 'Leido'),
(19, 20, 'Será sobre historia universal.', '2025-09-20 04:00', 'Leido');
```

-- Consultas rápidas de todas las tablas principales.

```
SELECT * FROM Usuarios;
SELECT * FROM Perfil;
SELECT * FROM Contenido;
SELECT * FROM Suscripcion;
SELECT * FROM Pago;
SELECT * FROM Mensaje;
```

De así quererlo también se puede visualizar el script en el siguiente enlace de [GitHub](#). En ese mismo repositorio también podrás ver el diagrama entidad relación.

Auditoría y Registro de Logs en la Base de Datos

Lógica de nuestros logs y auditoría

En nuestro proyecto educativo, implementamos una política rigurosa de **auditoría automatizada** utilizando las capacidades de Workbench, MySQL, MariaDB, XAMPP o Docker.

Los **logs** de la base de datos son generados y almacenados automáticamente por el motor, exclusivamente accesibles por los administradores o programadores autorizados.

Estos registros quedan fuera del acceso de los usuarios y no se visualizan ni manipulan desde la aplicación.

Los logs capturan aspectos fundamentales como:

- Fechas y horarios de las operaciones realizadas.
- Identidad del administrador o usuario técnico del motor.
- Tipo de instrucción ejecutada (INSERT, UPDATE, DELETE, ALTER, SELECT, etc.).
- Resultado y estado de los procesos, así como eventos y errores críticos.

Para garantizar la trazabilidad, incluimos en este mismo repositorio/documentación el **código SQL necesario para habilitar, ubicar y respaldar nuestros logs**, así como su lógica de recuperación y exportación. Esto hace que la política sea auditável y reproducible por cualquier responsable técnico.

¿Cómo gestionamos y resguardamos nuestros logs?

- Habilitamos el log general del servidor (y, si corresponde, el log binario transaccional).
 - Por ejemplo, en MySQL/MariaDB/Workbench:
 - SET GLOBAL general_log = 'ON';
 - SET GLOBAL general_log_file = '/ruta/nuestro_log_general.txt';
 -
- Solo los administradores o programadores extraen el archivo para resguardo:
 - Workbench: Server > Logs > View General Log File...
 - XAMPP/MariaDB/Docker: Copia directa desde el directorio de datos designado en la configuración.
- Antes de cada respaldo relevante y entrega del sistema, **incluimos el archivo de logs junto con el código del script** en la presentación, asegurando evidencia de la trazabilidad y programación aplicada.
- El log, como el respaldo, queda almacenado en .txt y puede ser inspeccionado solo por responsables técnicos, manteniendo la integridad y confidencialidad.

¿Cómo recuperamos nuestros logs?

- Solo administradores o programadores responsables accedemos al log:
 - Workbench: Server > Logs > View General Log File..., o exportación directa.
 - XAMPP/MariaDB/Docker: Abrimos el archivo definido en general_log_file.
- Para la presentación del sistema, copiamos el .txt relevante junto al script fuente y documentación, asegurando su resguardo y correcta identificación.

Política de respaldos de la base de datos

Nuestra política de respaldos asegura la integridad, disponibilidad y recuperabilidad de los datos, reduciendo los riesgos frente a fallos, errores o ataques.

Las operaciones de respaldo, restauración y resguardo están exclusivamente a cargo del **administrador o técnico** responsable del sistema, quien también controla el acceso a los logs.

Implementación del entorno con Docker

Implementamos el entorno de la base de datos utilizando Docker para garantizar portabilidad, persistencia y facilidad en las restauraciones y respaldos. El siguiente comando fue empleado para crear el contenedor MySQL:

```
docker run -d \
--name mysql_docker \
-e MYSQL_ROOT_PASSWORD=root \
-e MYSQL_DATABASE=basepruebasTC4 \
-p 3307:3306 \
-v C:\Users\user\docker-mysql\data:/var/lib/mysql \
mysql:8.0
```

- **-name mysql_docker:** Nombra el contenedor.
- **e MYSQL_ROOT_PASSWORD=root:** Contraseña para el usuario root.
- **e MYSQL_DATABASE=basepruebasTC4:** Base de datos inicial.
- **p 3307:3306:** Asocia el puerto local 3307 al interno 3306 del contenedor.
- **v:/var/lib/mysql:** Volumen persistente para datos.
- **mysql:8.0:** Imagen y versión de MySQL utilizada.

1. Ubicación y frecuencia

- Todos los respaldos completos y los logs se almacenan en **ubicaciones físicas externas** (discos externos, nube institucional, volúmenes Docker independientes, etc.).
- El técnico o administrador realiza como mínimo un **respaldo completo semanal** y otro previo a cada actualización importante.

2. Validación y restauración

- El técnico/administrador valida cada respaldo restaurándolo periódicamente en ambientes de prueba controlados (Workbench, XAMPP, Docker).
- Documenta cada respaldo, restauración y validación en los registros internos del entorno de administración.

3. Tipos de respaldo recomendados y ejemplos

Tipo	Descripción	Frecuencia sugerida
Completo	Copia completa de la base de datos	Semanal y antes de cambios
Diferencial	Solo los cambios desde el último respaldo completo	Diario

Ejemplo usando Workbench, MariaDB o XAMPP:

- Respaldo completo (en terminal o consola Docker):
 - mysqldump -u lumen_user -p Project > respaldo_bd_project_2025_11_15.sql
 -
 - Restauración:
 - mysql -u lumen_user -p Project < respaldo_bd_project_2025_11_15.sql
 -
 - En Workbench: ir a Server > Data Export O Server > Data Import.
 - **Incluimos siempre los archivos de logs y respaldos junto con el código del script y la documentación técnica, para facilitar auditoría y trazabilidad.**

4. Buenas prácticas y políticas internas

- Nunca almacenamos todos los respaldos y logs en el mismo servidor físico.
- Protegemos los archivos de respaldo y logs frente a accesos no autorizados, conforme a las mejores prácticas técnicas.
- Registramos toda restauración y respaldo relevante en la bitácora interna administrada por el técnico o administrador del proyecto/base.

Hoja testigo:

Firma del Profesor