

PRESENTACIÓN PROYECTO BBDD SQL

MI ESTILO

CODERHOUSE

Enero 2024



Sobre MI ESTILO

Mi Estilo es una red digital de generación de contenido, servicios de coaching, agencia de talentos, creatividad y producción audiovisual para público femenino con edad entre 18 y 35 años, interesadas en la moda, la belleza, el estilo de vida y el emprendimiento.



Brief

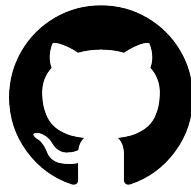
Nombre del proyecto: Mi Estilo

Idea de negocio y tipo: B2C

Oferta de mercado (producto/servicio): Red digital de generación de contenido (website/blog), servicios de coaching, agencia de talentos, creatividad y producción audiovisual para mujeres de 18 a 35 años, interesadas en la moda, la belleza, el estilo de vida y el emprendimiento.

Target: Mujeres de 18 a 35 años, interesadas en la moda, la belleza, el estilo de vida y el emprendimiento.

Necesidad o problema de los usuarios: Las mujeres de 18 a 35 años buscan información y recursos para desarrollar su estilo personal, mejorar su autoestima y alcanzar sus objetivos profesionales y personales. Sin embargo, a menudo encuentran esta información dispersa y fragmentada en diferentes canales.



PRESENTACIÓN PROYECTO BBDD SQL

Mi Estilo



Proyecto BBDD SQL

En este proyecto, se diseñará una base de datos SQL para un marketplace de moda, belleza y estilo de vida llamado Mi Estilo. El marketplace está diseñado para facilitar la compra y venta de productos y servicios entre usuarios. El marketplace permite a los vendedores publicar anuncios de productos o servicios que deseen vender, y a los compradores buscar artículos de su interés. Los usuarios pueden realizar transacciones en una variedad de categorías, desde ropa y accesorios hasta maquillaje y tratamientos de belleza.

Descripción de la Situación de Negocio

Mi Estilo está dirigido a mujeres de 18 a 35 años interesadas en moda, belleza y estilo de vida. El marketplace ofrece una amplia gama de productos y servicios, desde ropa y accesorios de marcas reconocidas hasta tratamientos de belleza personalizados.

El marketplace está creciendo rápidamente y se espera que continúe creciendo en los próximos años. Esto plantea desafíos en términos de gestión eficiente de la información, seguridad de los datos, realización de transacciones, confiabilidad y transparencia, y escalabilidad.

Base de datos: Objetivo

Objetivo de la Base de Datos

La base de datos debería cumplir con los siguientes objetivos:

- **Gestión Eficiente de la Información:** La base de datos debe permitir almacenar y acceder a la información de manera eficiente, incluso a medida que el marketplace crece.
- **Seguridad de Datos:** La base de datos debe garantizar la protección de la información confidencial de los usuarios, como datos personales y financieros.
- **Facilitación de Transacciones:** La base de datos debe facilitar la realización de transacciones, asegurando que los usuarios puedan buscar productos, realizar compras y gestionar sus ventas de manera efectiva.
- **Confiabilidad y Transparencia:** La base de datos debe fomentar la confiabilidad entre los usuarios mediante la gestión de calificaciones y reseñas, lo que ayudará a tomar decisiones informadas al comprar o vender productos y servicios.
- **Escalabilidad:** La base de datos debe ser escalable para adaptarse al crecimiento del marketplace a lo largo del tiempo sin comprometer el rendimiento.

Base de datos: Objetivo

Entidades y Relaciones

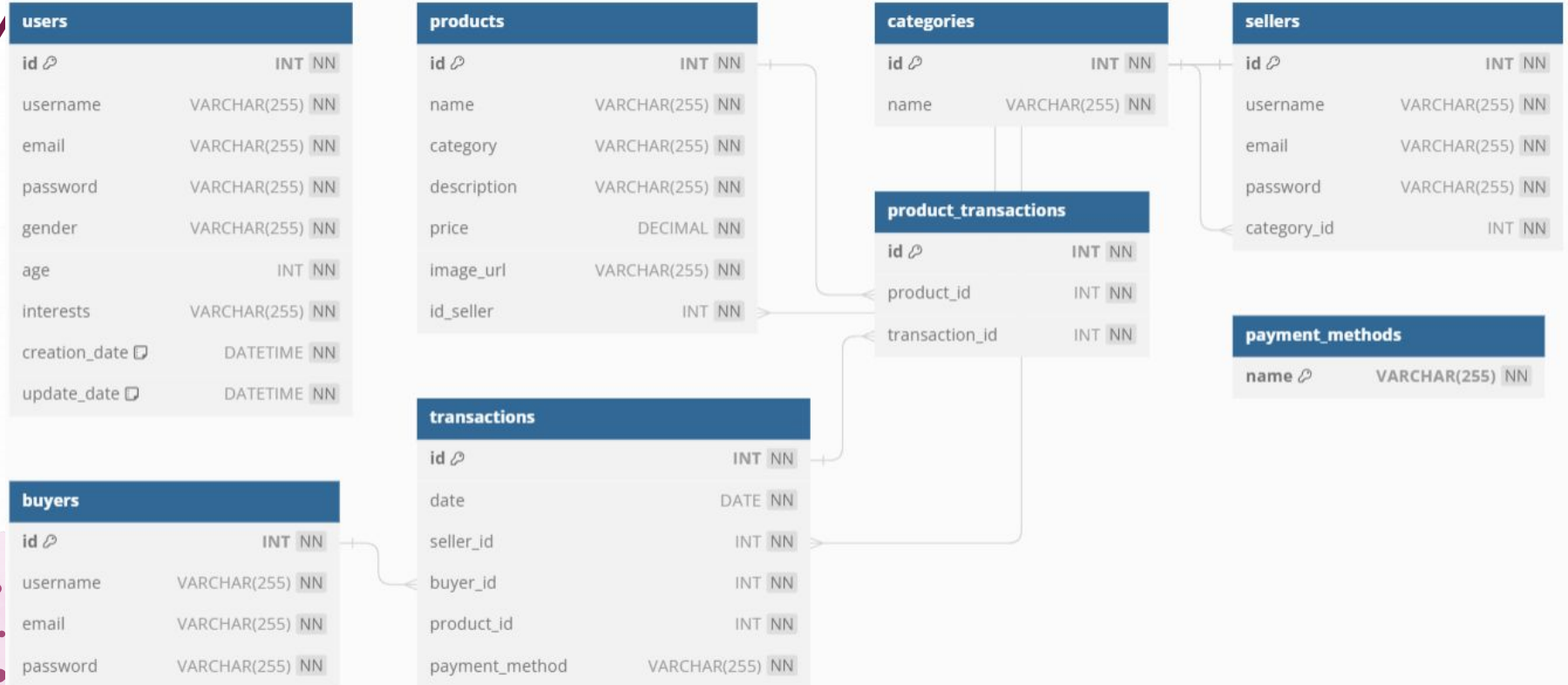
Las entidades de la base de datos son las siguientes:

- **Usuario:** Representa a un usuario registrado en el marketplace.
- **Producto:** Representa un producto o servicio disponible para la venta en el marketplace.
- **Transacción:** Representa una transacción realizada entre un vendedor y un comprador en el marketplace.
- **Calificación:** Representa una calificación otorgada por un usuario a otro usuario.
- **Reseña:** Representa una reseña escrita por un usuario sobre un producto o servicio.

Las relaciones entre estas entidades son las siguientes:

- **Usuario:** Un usuario puede crear uno o varios productos.
- **Producto:** Un producto puede ser vendido por uno o varios usuarios.
- **Transacción:** Una transacción involucra a un vendedor y un comprador.
- **Usuario:** Un usuario puede otorgar una calificación a otro usuario.
- **Producto:** Un usuario puede escribir una reseña sobre un producto.

Diagrama Entidad-Relación



Descripción de tablas

La tabla users almacena información sobre los usuarios registrados en el marketplace Mi Estilo. Los campos de la tabla incluyen:

id:	El	identificador	único	del	usuario.
name:	El	nombre		del	usuario.
e-mail:	La	dirección	de	correo electrónico	del usuario.
password:	La	contraseña		del	usuario.
gender:	El	género		del	usuario.
age:	La	edad		del	usuario.
interests:	Los	intereses		del	usuario.
notes:	Notas adicionales sobre los intereses del usuario.				

Descripción de tablas

La tabla products almacena información sobre los productos que se ofrecen en el marketplace Mi Estilo. En este caso serían los talleres y webinars. Los campos de la tabla incluyen:

id:	El	identificador	único	del	producto.
name:	El	nombre		del	producto.
description:	Una	descripción		del	producto.
price:	El	precio		del	producto.
category:	La	categoría		del	producto.
coach:	El autor del producto.				

Descripción de tablas

Tabla	Users						
Descripción	La tabla users almacena información sobre los usuarios registrados en el marketplace Mi Estilo.						
KEY	COLUMN	TYPE	LENGTH	NOT NULL	UNIQUE	DEFAULT	NOTES
id	id	integer	11	YES	YES	NULL	
name	name	varchar	255	YES	NO	NULL	
email	email	varchar	255	YES	NO	NULL	
password	password	varchar	255	YES	NO	NULL	
gender	gender	varchar	25	NO	NO	NULL	
age	age	integer	11	NO	NO	NULL	
interests	interests	text	255	NO	NO	NULL	
created_at	created_at	timestamp	19	YES	NO	NULL	
updated_at	updated_at	timestamp	19	YES	NO	NULL	

Descripción de tablas

Tabla	Products						
Descripción	La tabla products almacena información sobre los productos que se ofrecen en el marketplace Mi Estilo.						
KEY	COLUMN	TYPE	LENGTH	NOT NULL	UNIQUE	DEFAULT	NOTES
id	id	integer	11	YES	YES	NULL	
name	name	varchar	255	YES	NO	NULL	
description	description	text	255	YES	NO	NULL	
price	price	decimal	10,2	YES	NO	NULL	
category	category	varchar	255	YES	NO	NULL	
coach	coach	varchar	255	YES	NO	NULL	
created_at	created_at	timestamp	19	YES	NO	NULL	
updated_at	updated_at	timestamp	19	YES	NO	NULL	

Descripción de tablas

Tabla	Categories						
Descripción	La tabla products almacena información sobre los productos que se ofrecen en el marketplace Mi Estilo.						
KEY	COLUMN	TYPE	LENGTH	NOT NULL	UNIQUE	DEFAULT	NOTES
id	id	INT	11	YES	NO	NULL	Primary key
name	name	VARCHAR	255	YES	YES	NULL	Category name

Tabla	Sellers						
Descripción	La tabla products almacena información sobre los vendedores que comercializan sus webinars/talleres en el marketplace de Mi Estilo.						
KEY	COLUMN	TYPE	LENGTH	NOT NULL	UNIQUE	DEFAULT	NOTES
id	id	INT	11	YES	NO	NULL	Primary key
username	username	VARCHAR	255	YES	YES	NULL	Seller's username
email	email	VARCHAR	255	YES	YES	NULL	Seller's email address
password	password	VARCHAR	255	YES	YES	NULL	Seller's password
category_id	category_id	INT	11	YES	NO	NULL	Foreign key referencing the category

Descripción de tablas

Tabla	Buyers						
Descripción	La tabla products almacena información sobre los clientes/compradores.						
KEY	COLUMN	TYPE	LENGTH	NOT NULL	UNIQUE	DEFAULT	NOTES
id	id	INT	11	YES	NO	NULL	Primary key
username	username	VARCHAR	255	YES	YES	NULL	Buyer's username
email	email	VARCHAR	255	YES	YES	NULL	Buyer's email address
password	password	VARCHAR	255	YES	YES	NULL	Buyer's password

Tabla	Transactions						
Descripción	La tabla products almacena información sobre las transacciones realizadas en el marketplace de Mi Estilo.						
KEY	COLUMN	TYPE	LENGTH	NOT NULL	UNIQUE	DEFAULT	NOTES
id	id	INT	11	YES	NO	NULL	Primary key
date	date	DATE	10	YES	YES	NULL	Transaction date
seller_id	seller_id	INT	11	YES	NO	NULL	Foreign key referencing the seller
buyer_id	buyer_id	INT	11	YES	NO	NULL	Foreign key referencing the buyer
product_id	product_id	INT	11	YES	NO	NULL	Foreign key referencing the product
payment_method	payment_method	VAR	255	YES	YES	NULL	Payment method used for the transaction

Descripción de tablas

Tabla	Product transactions						
Descripción	La tabla products almacena información sobre los productos vendidos/comprados en el marketplace de Mi Estilo.						
KEY	COLUMN	TYPE	LENGTH	NOT NULL	UNIQUE	DEFAULT	NOTES
id	id	INT	11	YES	NO	NULL	Primary key
product_id	product_id	INT	11	YES	NO	NULL	Foreign key referencing the product
transaction_id	transaction_id	INT	11	YES	NO	NULL	Foreign key referencing the transaction

	Payment Methods						
Descripción	La tabla products almacena información sobre los métodos de pago utilizados en el marketplace de Mi Estilo.						
KEY	COLUMN	TYPE	LENGTH	NOT NULL	UNIQUE	DEFAULT	NOTES
name	name	VARCHAR	255	YES	YES	NULL	Payment method name

Creación de tablas

Users, Products, Categories

```
Table "users" {  
  id INT [pk, not null, increment]  
  username VARCHAR(255) [unique, not null]  
  email VARCHAR(255) [not null]  
  password VARCHAR(255) [not null]  
  gender VARCHAR(255) [not null]  
  age INT [not null]  
  interests VARCHAR(255) [not null]  
  creation_date DATETIME [not null, default:  
    `CURRENT_TIMESTAMP`]  
  update_date DATETIME [not null, default:  
    `CURRENT_TIMESTAMP`]  
}
```

```
Table "products" {  
  id INT [pk, not null, increment]  
  name VARCHAR(255) [not null]  
  category VARCHAR(255) [not null]  
  description VARCHAR(255) [not null]  
  price DECIMAL [not null]  
  image_url VARCHAR(255) [not null]  
  id_seller INT [not null]  
}
```

```
Table "categories" {  
  id INT [pk, not null, increment]  
  name VARCHAR(255) [not null]  
}
```


Creación de tablas

Transactions, Sellers, Buyers, Payments, Product transactions

```
Table "transactions" {  
  id INT [pk, not null, increment]  
  date DATE [not null]  
  seller_id INT [not null]  
  buyer_id INT [not null]  
  product_id INT [not null]  
  payment_method VARCHAR(255) [not null]  
}
```

```
Table "sellers" {  
  id INT [pk, not null, increment]  
  username VARCHAR(255) [not null]  
  email VARCHAR(255) [not null]  
  password VARCHAR(255) [not null]  
  category_id INT [not null]  
}
```

```
Table "buyers" {  
  id INT [pk, not null, increment]  
  username VARCHAR(255) [not null]  
  email VARCHAR(255) [not null]  
  password VARCHAR(255) [not null]  
}
```

```
Table "payment_methods" {  
  name VARCHAR(255) [pk, not null]  
}
```

```
Table "product_transactions" {  
  id INT [pk, not null, increment]  
  product_id INT [not null]  
  transaction_id INT [not null]  
}
```

Vistas

Best selling course & Top seller

```
CREATE VIEW top_5_best_selling_courses AS
SELECT
c.id,
c.name,
c.description,
c.price,
c.created_at,
c.updated_at,
c.category_id,
COUNT(*) AS number_of_sales
FROM product_transactions pt
INNER JOIN products c ON pt.product_id = c.id
GROUP BY c.id
ORDER BY number_of_sales DESC
LIMIT 5;
```

```
CREATE VIEW top_seller AS
SELECT
c.seller_id,
u.username,
COUNT(*) AS number_of_sales,
(
SELECT COUNT(*)
FROM product_transactions pt
INNER JOIN products c ON pt.product_id = c.id
WHERE c.seller_id = pt.seller_id
) AS total_courses_sold
FROM product_transactions pt
INNER JOIN products c ON pt.product_id = c.id
INNER JOIN users u ON c.seller_id = u.id
GROUP BY c.seller_id
ORDER BY number_of_sales DESC
LIMIT 1;
```

Funciones

Función

Calcular el promedio de edad de Users registrados en Mi Estilo.

```
CREATE FUNCTION get_average_age_of_registered_users()  
RETURNS DECIMAL(10,2) AS  
BEGIN  
RETURN (  
    SELECT AVG(age)  
FROM users  
);  
END;
```

```
CREATE FUNCTION get_courses_by_seller(seller_id INT)
RETURNS TABLE (
  id INT,
  name VARCHAR(255),
  description VARCHAR(255),
  price DECIMAL(10,2),
  created_at DATETIME,
  updated_at DATETIME
) AS
BEGIN
  RETURN (
    SELECT
      c.id,
      c.name,
      c.description,
      c.price,
      c.created_at,
      c.updated_at
    FROM products c
    WHERE c.seller_id = seller_id
  );
END;
```

Función

Curso más vendido por seller ID

Stored Procedures

Utilidad

Si tenemos una tabla transactions con un millón de registros. Al ejecutar el procedimiento almacenado get_total_transactions sin ningún cambio, la consulta SELECT COUNT(*) se ejecutará un millón de veces. Esto puede ser muy ineficiente, especialmente si la tabla transactions es grande.

Para mejorar la eficiencia del procedimiento almacenado, podemos utilizar una variable para almacenar el número de registros. Esto evitará que la consulta SELECT se ejecute más de una vez.

```
CREATE PROCEDURE get_total_transactions(  
  IN table_name VARCHAR(255) = 'transactions',  
  IN order_type VARCHAR(255)  
)  
;  
BEGIN  
  DECLARE total_transactions INT;  
  CREATE TEMPORARY TABLE temp_results (total_transactions INT);  
  IF order_type = 'ASC' THEN  
    INSERT INTO temp_results  
    SELECT COUNT(*)  
    FROM table_name  
    ORDER BY date ASC;  
  ELSE  
    INSERT INTO temp_results  
    SELECT COUNT(*)  
    FROM table_name  
    ORDER BY date DESC;  
  END IF;  
  SELECT * FROM temp_results;  
  DROP TABLE temp_results;  
END;
```

Triggers

Products_log

```
CREATE TABLE products_log (  
  id INT AUTO_INCREMENT,  
  product_id INT,  
  user_id INT,  
  action VARCHAR(255),  
  date DATETIME,  
  time TIME,  
  PRIMARY KEY (id)  
);
```

Utilidad

Registrar las acciones que se realizan en la tabla Products. Esto permite realizar un seguimiento de las modificaciones realizadas en los datos, y puede ser útil para fines de auditoría o para generar informes mensuales de Mi Estilo.

Triggers

```
CREATE TRIGGER products_log_before_update
BEFORE UPDATE ON products
FOR EACH ROW
BEGIN
    INSERT INTO products_log (product_id, user_id, action, date, time)
    VALUES (OLD.id, OLD.user_id, 'UPDATE', NOW(), NOW());
END;

PRIMARY KEY (id)
);
```

```
CREATE TRIGGER products_log_after_update
AFTER UPDATE ON products
FOR EACH ROW
BEGIN
    INSERT INTO products_log (product_id, user_id, action, date, time)
    VALUES (NEW.id, NEW.user_id, 'UPDATE', NOW(), NOW());
END;
```

Triggers

Product_transactions_log

```
CREATE TABLE
product_transactions_log (
id INT AUTO_INCREMENT,
product_transaction_id INT,
user_id INT,
action VARCHAR(255),
date DATETIME,
time TIME,
PRIMARY KEY (id)
);
```

Utilidad

Registrar las acciones que se realizan en la tabla Product_transactions. Esto permite realizar un seguimiento y registro de los Users que realizaron una operación, la fecha y la hora.

Triggers

```
CREATE TRIGGER product_transactions_log_before_insert
BEFORE INSERT ON product_transactions
FOR EACH ROW
BEGIN
    INSERT INTO product_transactions_log (product_transaction_id, user_id, action, date, time)
    VALUES (NEW.id, NEW.user_id, 'INSERT', NOW(), NOW());
END;

PRIMARY KEY (id)
);
```

```
CREATE TRIGGER product_transactions_log_after_insert
AFTER INSERT ON product_transactions
FOR EACH ROW
BEGIN
    INSERT INTO product_transactions_log (product_transaction_id, user_id, action,
    date, time)
    VALUES (NEW.id, NEW.user_id, 'UPDATE', NOW(), NOW());
END;
```



GRACIAS



Fonts & colors used

This presentation has been made using the following fonts:

Overpass

(<https://fonts.google.com/specimen/Overpass>)

Open Sans

(<https://fonts.google.com/specimen/Open+Sans>)

#630742

#c63d9d

#dc6fb8

#e89fdc

#dc9fe8

#7dc4b2

#a8d7cd