



M I D T E R M

B D

YASMIN NERI
AGOSTO, 2024

INDICE DE CONTENIDOS

- 
1. DESAROLLADORA
 2. EL PROYECTO
 3. RUTAS
 4. DIAGRAMA ER
 5. FUNCIONALIDADES
WEB
 6. TECNOLOGIAS

DESARROLADORA
DEL PROYECTO

DESARROLADORA

Yasmin Neri, 29 años (casi 30)
Graduada en fisioterapia
Estudiante de ingeniería de software
Experiencia en C, Git y Github
Trabajo en equipo

The background features a dark, slightly blurred image of a computer keyboard at the bottom. Above the keyboard, a portion of a code editor window is visible, displaying lines of HTML, CSS, and JavaScript code. The code includes references to Bootstrap, a video player, and social media links. The overall aesthetic is professional and tech-oriented.

EL PROYECTO



BASE DE DATOS

FRONTEND

BACKEND

OBJETIVO:

Desarrollar y gestionar una base de datos con múltiples requisitos, integrando relaciones complejas, búsquedas avanzadas, triggers, transacciones y funciones, presentando el trabajo mediante una interfaz web interactiva.



Transforma tu espacio
estilo

Conoce a nuestros productos

Explorar

PROYECTO:

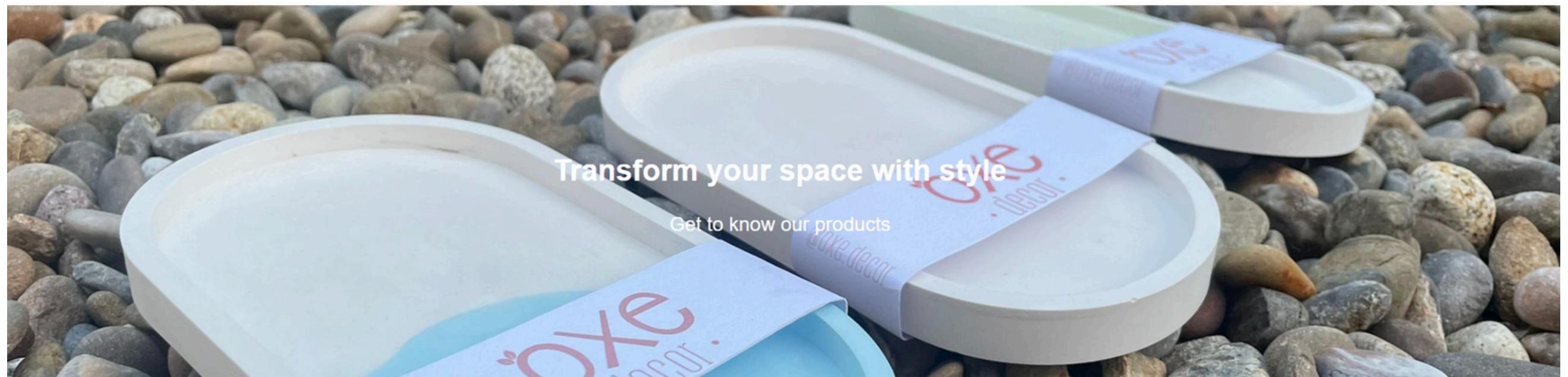
REGISTRO



[Home](#) [Products](#) [Register](#)



CARRO DE LA COMPRA



AÑADIR PRODUCTO

RUTAS HABILITADAS Y SUS FUNCIONES

RUTAS

```
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        username = request.form['username']
        address = request.form['address']
        phone = request.form['phone']
        email = request.form['email']
        password = request.form['password']

        conn = mysql.connector.connect(**db_config)
        cursor = conn.cursor()
        query = "INSERT INTO customers (name, address, phone, email) VALUES (%s, %s, %s, %s)"
        try:
            cursor.execute(query, (username, address, phone, email))
            conn.commit()
        except:
            conn.rollback()
        cursor.close()
        return redirect(url_for('success'))

    return render_template('register.html')

@app.route('/success')
def success():
    return "Registration successful =D"
```

RUTAS INCOMPLETAS

```
@app.route('/add_product', methods=['POST'])
def add_product():
    product = request.form['product']
    color = request.form['color']

    conn = mysql.connector.connect(**db_config)
    cursor = conn.cursor()

    query = "INSERT INTO new_product (product, color)) VALUES (%s, %s)"
    cursor.execute(query, (product, color))

    conn.commit()
    cursor.close()
    conn.close()

    return redirect('/')
```

RUTAS INCOMPLETAS

```
@app.route('/cart')
def cart():
    conn = mysql.connector.connect(**db_config)
    cursor = conn.cursor()

    query = """
        SELECT ci.id, p.name, p.price, ci.quantity
        FROM cart_items ci
        JOIN products p ON ci.product_id = p.id
    """
    cursor.execute(query)
    cart_items = cursor.fetchall()
    cursor.close()
    conn.close()
    return render_template('cart.html', cart_items=cart_items)

@app.route('/cart/add/<int:product_id>', methods=['POST'])
def add_to_cart(product_id):
    conn = mysql.connector.connect(**db_config)
    cursor = conn.cursor()

    query = "INSERT INTO cart_items (product_id, quantity) VALUES (%s, %s)"
    cursor.execute(query, (product_id, 1))
    conn.commit()
    cursor.close()
    conn.close()
    return redirect(url_for('cart'))
```

RUTAS INCOMPLETAS

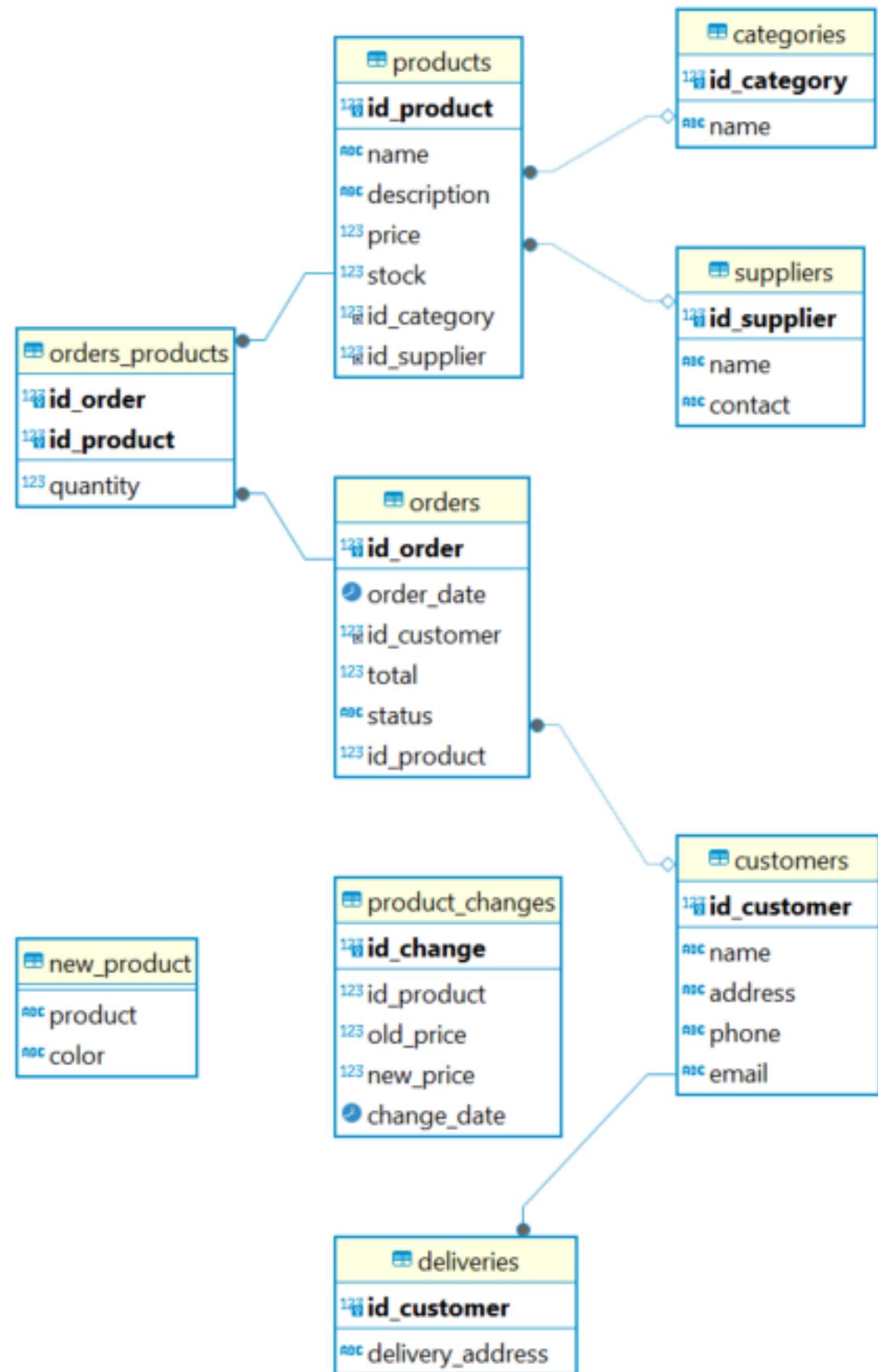
```
@app.route('/cart/remove/<int:item_id>', methods=['POST'])
def remove_from_cart(item_id):
    conn = mysql.connector.connect(**db_config)
    cursor = conn.cursor()

    query = "DELETE FROM cart_items WHERE id = %s"
    cursor.execute(query, (item_id,))
    conn.commit()
    cursor.close()
    conn.close()
    return redirect(url_for('cart'))
```

DIAGRAMA ER

Propiedades

Diagrama ER



FUNCIONALIDADES DE LA WEB



Home



Name:

Address:

Phone:

E-mail:

Password:

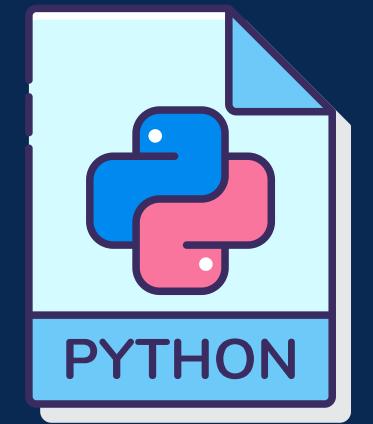
Submit

Products list

Find product

Haven't you found the product you like? Tell us a product you would like to see in our catalog:

Product Color:



ChatGPT

TECNOLOGIAS UTILIZADAS



git



GESTIÓN DE TIEMPO

```
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="container">
          <Title name="our" title="product">
            <div className="row">
              <ProductConsumer>
                {(value) => {
                  |   |   |   console.log(value)
                  |   |   |
                }}
              </ProductConsumer>
            </div>
          </div>
        </div>
      </React.Fragment>
```

MUCHAS GRACIAS

YASMIN NERI
YASMIINERI@GMAIL.COM

```
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="container">
          <Title name="our" title="product">
            <div className="row">
              <ProductConsumer>
                {(value) => {
                  |   |   |   console.log(value)
                  |   |   |
                  |   |   |   }}
                </ProductConsumer>
              </div>
            </div>
          </div>
        </React.Fragment>
```