



UNIVERSIDAD BANCARIA DE MÉXICO

"*Constancia Unidad y trabajo*"

INGENIERÍA EN SISTEMAS COMPUTACIONALES

RECONOCIMIENTO DE VALIDEZ OFICIAL DE ESTUDIOS DE LA
SECRETARÍA DE EDUCACIÓN PÚBLICA No. 2022241 DE FECHA 13
DE SEPTIEMBRE DE 2002.

PROYECTO CARRITO DE COMPRAS
DESARROLLO DE APLICACIONES PARA INTERNET AVANZADO 3

PROFESOR:

ALEXIS BOCHE HERNÁNDEZ

GRUPO:

ISC 1002 V

ALUMNO:
FERNANDO URIEL LEOCADIO RODRIGUEZ



Índice

No se encontraron entradas de tabla de contenido.

Introducción

LEGO STORE

Productos Carrito (0) Login Registro

Catálogo de LEGO



LEGO Star Wars Millennium Falcon
\$169.99

Agregar al Carrito



LEGO Harry Potter Hogwarts Castle
\$399.99

Agregar al Carrito



LEGO Technic Bugatti Chiron
\$349.99

Agregar al Carrito



LEGO City Police Station
\$99.99

Agregar al Carrito



LEGO STORE

Productos Carrito (0) Login Registro

Tu Carrito

El carrito está vacío.

Iniciar Sesión

Usuario

Contraseña

Entrar**Registrarse**

Elige un usuario

Elige contraseña

Registrar

Tu Carrito

Producto	Precio	Cantidad	Subtotal	Acción
LEGO Star Wars Millennium Falcon	\$169.99	- 1 +	\$169.99	Borrar
LEGO Harry Potter Hogwarts Castle	\$399.99	- 1 +	\$399.99	Borrar
LEGO Technic Bugatti Chiron	\$349.99	- 1 +	\$349.99	Borrar

Total: \$919.97

Debes iniciar sesión para comprar.

[Ir a Login](#)

Iniciar Sesión

```

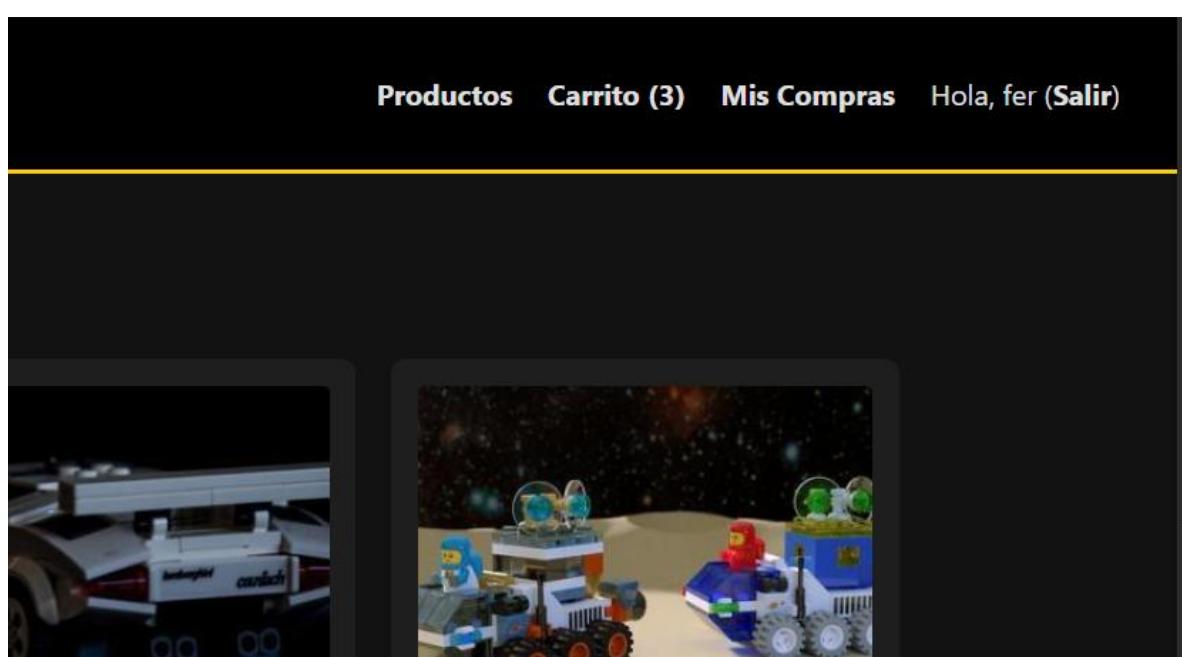
1 •  DROP DATABASE IF EXISTS lego_store;
2
3 •  CREATE DATABASE lego_store;
4 •  USE lego_store;
5
6    -- Tabla de Usuarios
7 •  CREATE TABLE users (
8      id INT AUTO_INCREMENT PRIMARY KEY,
9      username VARCHAR(50) NOT NULL UNIQUE,
10     password VARCHAR(255) NOT NULL, -- En producción usar hash
11     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
12   );
13
14 •  select * from users;
15
16    -- Tabla de Productos
17 •  CREATE TABLE products (
18      id INT AUTO_INCREMENT PRIMARY KEY,
19      name VARCHAR(100) NOT NULL,
20      price DECIMAL(10, 2) NOT NULL,
21      image_url VARCHAR(255) NOT NULL
22   );
23
24    -- Tabla de Ordenes (Cabecera)
25 •  CREATE TABLE orders (
26      id INT AUTO_INCREMENT PRIMARY KEY,
27      user_id INT NOT NULL,
28      total DECIMAL(10, 2) NOT NULL,
29      date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
30      FOREIGN KEY (user_id) REFERENCES users(id)
31   );
32
33    -- Detalle de Ordenes
34 •  CREATE TABLE order_items (
35      id INT AUTO_INCREMENT PRIMARY KEY,
36      order_id INT NOT NULL,
37      product_name VARCHAR(100),
38      quantity INT NOT NULL,
39      price DECIMAL(10, 2) NOT NULL,
40      FOREIGN KEY (order_id) REFERENCES orders(id)
41   );
42
43 •  CREATE TABLE pending_carts (
44      user_id INT PRIMARY KEY,
45      cart_data JSON NOT NULL, -- Almacena el carrito completo (array de objetos) en formato JSON
46      FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
47   );
48
49    -- Insertar 20 Productos de LEGO (URLs genéricas que cargan imágenes de Lego)
50 •  INSERT INTO products (name, price, image_url) VALUES
51  ('LEGO Star Wars Millennium Falcon', 169.99, 'https://loremflickr.com/300/300/lego,starwars?random=1'),
52  ('LEGO Harry Potter Hogwarts Castle', 399.99, 'https://loremflickr.com/300/300/lego,castle?random=2'),
53  ('LEGO Technic Bugatti Chiron', 349.99, 'https://loremflickr.com/300/300/lego,car?random=3'),
54  ('LEGO City Police Station', 99.99, 'https://loremflickr.com/300/300/lego,city?random=4'),
55  ('LEGO Architecture Statue of Liberty', 119.99, 'https://loremflickr.com/300/300/lego,architecture?random=5'),
56  ('LEGO Creator Expert Roller Coaster', 379.99, 'https://loremflickr.com/300/300/lego,park?random=6'),
57  ('LEGO Ninjago City Gardens', 299.99, 'https://loremflickr.com/300/300/lego,ninja?random=7'),
58  ('LEGO Marvel Infinity Gauntlet', 69.99, 'https://loremflickr.com/300/300/lego,marvel?random=8'),
59  ('LEGO Ideas Tree House', 199.99, 'https://loremflickr.com/300/300/lego,tree?random=9'),
60  ('LEGO Super Mario Starter Course', 59.99, 'https://loremflickr.com/300/300/lego,mario?random=10'),
61  ('LEGO Technic Lamborghini Sián', 379.99, 'https://loremflickr.com/300/300/lego,lambo?random=11'),
62  ('LEGO Friends Apartment', 149.99, 'https://loremflickr.com/300/300/lego,friends?random=12'),
63  ('LEGO Batman Batmobile', 249.99, 'https://loremflickr.com/300/300/lego,batman?random=13'),
64  ('LEGO Jurassic Park T-Rex', 249.99, 'https://loremflickr.com/300/300/lego,dino?random=14'),
65  ('LEGO Art World Map', 249.99, 'https://loremflickr.com/300/300/lego,map?random=15'),

```

```
14 • select * from users;
```

```
15
```

	id	username	password	created_at
▶	1	fer	1234	2025-11-28 12:53:35
2	admin	1234	2025-11-28 12:53:50	
*	NULL	NULL	NULL	NULL



LEGO STORE

Productos Carrito (3) Mis Compras Hola, fer (Salir)

Tu Carrito

Producto	Precio	Cantidad	Subtotal	Acción
LEGO Star Wars Millennium Falcon	\$169.99	<input type="button" value="-"/> 1 <input type="button" value="+"/>	\$169.99	Borrar
LEGO Harry Potter Hogwarts Castle	\$399.99	<input type="button" value="-"/> 1 <input type="button" value="+"/>	\$399.99	Borrar
LEGO Technic Bugatti Chiron	\$349.99	<input type="button" value="-"/> 1 <input type="button" value="+"/>	\$349.99	Borrar

Total: \$919.97

[Finalizar Compra y Descargar Ticket](#)

Traductor de Google G2A.COM - Open th... ChatGPT Proyectos sencillos... SQL Tutorial Bing Base44 ENEF Descargas ticket_4.pdf Abrir archivo Ver más

LEGO STORE

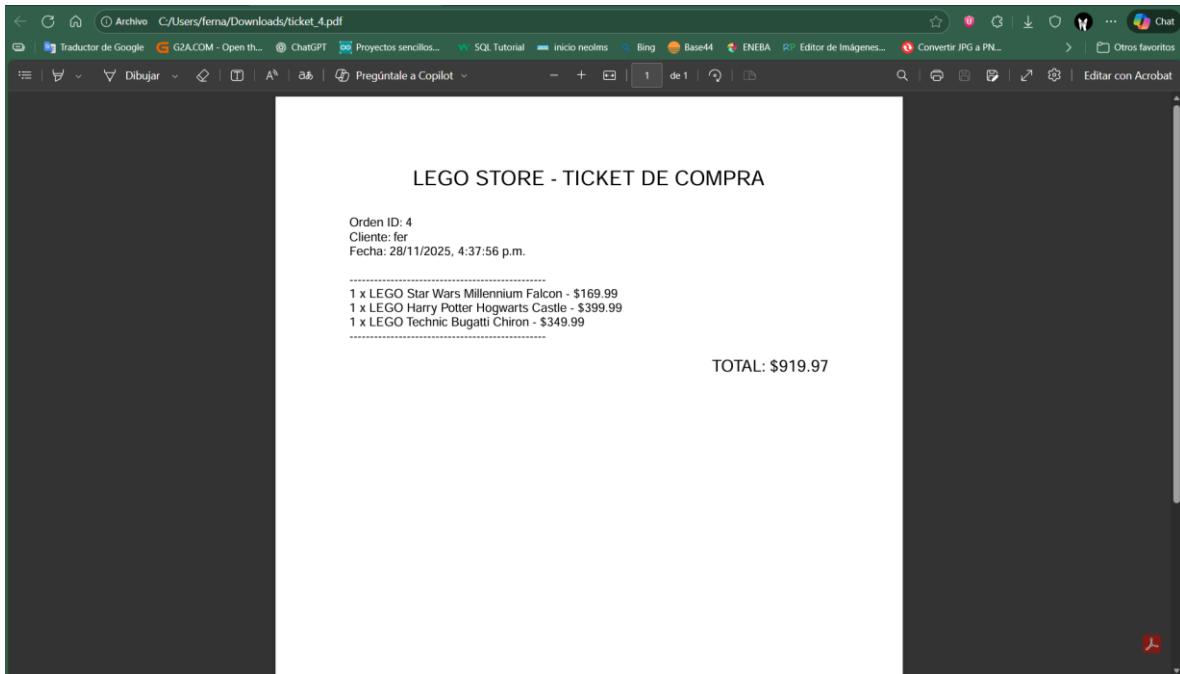
Productos Carrito (3) Mis Compras Hola, fer (Salir)

Tu Carrito

Producto	Precio	Cantidad	Subtotal	Acción
LEGO Star Wars Millennium Falcon	\$169.99	<input type="button" value="-"/> 1 <input type="button" value="+"/>	\$169.99	Borrar
LEGO Harry Potter Hogwarts Castle	\$399.99	<input type="button" value="-"/> 1 <input type="button" value="+"/>	\$399.99	Borrar
LEGO Technic Bugatti Chiron	\$349.99	<input type="button" value="-"/> 1 <input type="button" value="+"/>	\$349.99	Borrar

Total: \$919.97

[Finalizar Compra y Descargar Ticket](#)



The screenshot shows the LEGO STORE website interface. At the top, there is a navigation bar with links for "Productos", "Carrito (0)", "Mis Compras", and a user greeting "Hola, fer (Salir)".

Below the navigation bar, there is a section titled "Historial de Compras" (Purchase History) which displays the following table:

ID Orden	Fecha	Productos	Total
#4	28/11/2025, 4:37:56 p.m.	LEGO Star Wars Millennium Falcon (1), LEGO Harry Potter Hogwarts Castle (1), LEGO Technic Bugatti Chiron (1)	\$919.97
#3	28/11/2025, 3:03:04 p.m.	LEGO Star Wars Millennium Falcon (1)	\$169.99
#2	28/11/2025, 12:54:25 p.m.	LEGO Star Wars Millennium Falcon (1), LEGO Harry Potter Hogwarts Castle (1), LEGO Technic Bugatti Chiron (1), LEGO Architecture Statue of Liberty (1), LEGO Ideas Tree House (1)	\$1239.95
#1	28/11/2025, 12:54:10 p.m.	LEGO Star Wars Millennium Falcon (1)	\$169.99

```
JS app.js      X | dapia3 > parcial_2 > proyecto > JS app.js > app.get('/history') callback > query
1  const express = require('express');
2  const session = require('express-session');
3  const path = require('path');
4  const db = require('../db');
5  const PDFDocument = require('pdfkit');
6
7  const app = express();
8
9  app.set('view engine', 'ejs');
10 app.use(express.static(path.join(__dirname, 'public')));
11 app.use(express.urlencoded({ extended: true }));
12 app.use(express.json());
13
14 app.use(session({
15   secret: 'lego_secret_key',
16   resave: false,
17   saveUninitialized: true
18 }));
19
20 app.use((req, res, next) => {
21   if (!req.session.cart) req.session.cart = [];
22   res.locals.user = req.session.user;
23   res.locals.cartCount = req.session.cart.reduce((acc, item) => acc + item.quantity, 0);
24   next();
25 });
26
27 app.get('/', (req, res) => {
28   db.query('SELECT * FROM products', (err, products) => {
29     if (err) throw err;
30     res.render('index', { products });
31   });
32 });
33
34 app.post('/add-to-cart', (req, res) => {
35   const { id, name, price, image } = req.body;
36   const existingItem = req.session.cart.find(item => item.id === id);
37
38   if (existingItem) {
39     existingItem.quantity++;
40   } else {
41     req.session.cart.push({
42       id, name, price: parseFloat(price), image, quantity: 1
43     });
44   }
45   res.redirect('/');
46 });
47
48 app.get('/cart', (req, res) => {
49   const total = req.session.cart.reduce((acc, item) => acc + (item.price * item.quantity), 0);
50   res.render('cart', { cart: req.session.cart, total });
51 });
```

```
> app.use(session({ ... }));
> app.use((req, res, next) => { ... });
> app.get('/', (req, res) => { ... });
> app.post('/add-to-cart', (req, res) => { ... });
> app.get('/cart', (req, res) => { ... });
> app.post('/update-cart', (req, res) => { ... });

app.get('/login', (req, res) => res.render('login'));
app.get('/register', (req, res) => res.render('register'));

> app.post('/register', (req, res) => { ... });

app.get('/login', (req, res) => res.render('login'));
app.get('/register', (req, res) => res.render('register'));

> app.post('/login', (req, res) => { ... });

> app.get('/logout', (req, res) => { ... });

> app.get('/checkout', (req, res) => { ... });
> app.get('/history', (req, res) => { ... });

app.listen(3000, () => {
  console.log('Servidor corriendo en http://localhost:3000');
});
```

```
<%- include('header') %>
<h2>Catálogo de LEGO</h2>
<div class="product-grid">
  <% products.forEach(product => { %>
    <div class="card">
      ">
      <h3><%= product.name %</h3>
      <p class="price">$<%= product.price %</p>
      <form action="/add-to-cart" method="POST">
        <input type="hidden" name="id" value="<%= product.id %>">
        <input type="hidden" name="name" value="<%= product.name %>">
        <input type="hidden" name="price" value="<%= product.price %>">
        <input type="hidden" name="image" value="<%= product.image_url %>">
        <button type="submit">Agregar al Carrito</button>
      </form>
    </div>
  <% }) %>
</div>
<%- include('footer') %>
```

Conclusión