

PROYECTO FINAL
VALENTINE'S HOUSE – DESSERT SHOP

VALENTINA MAZORRA MEDINA 20201189730
SARA CRISTINA PRADA MEDINA 20201186762

PROF. JUAN ANTONIO CASTRO SILVA

PROGRAMACIÓN WEB
INGENIERÍA DE SOFTWARE
FACULTAD DE INGENIERÍA
UNIVERSIDAD SURCOLOMBIANA

2021-2

INTRODUCCIÓN

Con el transcurso del tiempo, las sociedades en todo el mundo experimentan transformaciones económicas que modifican la forma en cómo el mercado se comporta; y esto, a su vez, conlleva a cambiar el método con el que los empresarios y emprendedores desarrollan e implementan sus negocios, además de cambiar el modo en cómo los consumidores deciden interactuar con dichos negocios y acceder a sus productos y servicios.

Una de esas transformaciones se debió gracias a la tercera revolución industrial, cuyas bases se establecieron a partir de la tecnología de la información y la comunicación, pues esto dio paso a nuevos medios de interacción y gestión que alcanzó no solo el área industrial con relación a la automatización del proceso de generación de nuevas energías renovables, sino que también influyó el área de mercado.

En dicha área, se logran observar estos avances en la medida en que ya no solo existen tiendas físicas a dónde el consumidor interesado pueda dirigirse para adquirir un producto de su interés, sino que ahora también es posible encontrar y acceder a diferentes tipos de productos a través del internet por medio de las denominadas tiendas virtuales.

Para que este nuevo tipo de interacción vendedor-cliente (tienda virtual-cliente virtual) se experimente de forma amena y apropiada, son necesarios un mínimo de factores que aseguren el éxito de la transacción comercial originada entre estos entes. De forma resumida, los factores que podemos encontrar serían: una plataforma estable e intuitiva para el cliente o consumidor, información concisa y precisa de los productos ofrecidos en la plataforma así como los costos de cada uno, métodos claros de compra y adquisición de dichos productos y, protección y seguridad tanto para la plataforma como para el cliente contra bots, personas inescrupulosas o personas sin un fin más que ser maliciosos por diversión.

Con el fin de cumplir con todos estos requisitos, la creación, desarrollo e implementación de una tienda virtual está a cargo de profesionales relacionados al área de la ingeniería de software, la programación y afines.

Hoy en día, es muy común ver y utilizar tiendas virtuales gracias al extendido uso del internet y el avance tecnológico; sin embargo, las nuevas necesidades sociales y de mercado impulsan a, ya no solo crear tiendas virtuales, sino también a pulir y refinar los métodos que, siendo aplicados por programadores e ingenieros de software, permiten lograr tiendas virtuales cada vez más prácticas, funcionales y sofisticadas que antes.

Es en este contexto, y con el fin de formar a futuros profesionales ingenieros de software, que la Universidad Surcolombiana exige que estos adquieran las

herramientas básicas y la experiencia (aun si es parcial) para desarrollar páginas web y, más específicamente, tiendas virtuales. Es la clase de Programación Web la encargada de preparar a los estudiantes de ingeniería de software para hacer frente a este reto, así cómo es dónde se evaluará la forma en que los estudiantes fueron capaces de confrontar al susodicho reto mediante el desarrollo y presentación del trabajo final del curso, el cual corresponde a la creación de una tienda virtual en lenguaje Java y JavaScript.

Siguiendo con este lineamiento, el presente trabajo plantea la creación de una Repostería Virtual, en donde sea posible ver, elegir y, en teoría, comprar diferentes tipos de postres y productos de repostería en general.

PROBLEMA PROPUESTO

Implementación de una página web para una repostería, en la cual el cliente pueda filtrar los productos por categorías y comprarlos por medio de un sistema de carrito de compras. El cliente podrá añadir productos al carrito, modificar la cantidad de productos que sea o eliminar productos, también calcular el precio total.

OBJETIVO GENERAL

Desarrollar una tienda virtual intuitiva y cómoda en dónde sea posible vender y comprar productos de repostería.

OBJETIVOS ESPECÍFICOS

- Elegir los tipos de productos a ofrecer (categorías).
- Elegir los productos que se ofrecerán en la tienda virtual con base en los tipos o categorías seleccionadas.
- Definir entidades y atributos para desarrollar la base de datos de la tienda virtual.
- Generar el modelo entidad relación de la base de datos a implementar en la tienda virtual.
- Generar un proyecto en Eclipse para crear la tienda virtual.
- Generar la base de datos y conectarla al proyecto de la tienda virtual.
- Generar perfiles para los usuarios de la tienda virtual.
- Proteger los usuarios de la tienda virtual encriptando las contraseñas.
- Crear dominio de seguridad y configurar mecanismos de seguridad para los usuarios que accedan a la tienda virtual.
- Implementar un sistema de carrito de compras para procesar los pedidos de los clientes de la tienda virtual.
- Internacionalizar la tienda virtual.
- Elegir un framework Bootstrap acorde con la temática y necesidades de la tienda virtual planteada.
- Implementar, en el proyecto, el framework Bootstrap elegido.
- Estructurar el manejo de sesiones y cookies.

ARTÍCULOS OFRECIDOS

CATEGORÍA	PRODUCTOS
<i>Tortas</i>	Chocolate
	Napolitana
	Oreo
	Red Velvet
	Tres Leches
<i>Pies</i>	Arándano
	Cereza
	Durazno
	Frambuesa
	Mora
<i>Cheesecakes</i>	Arándano
	Chocolate
	Frutos Rojos
	Naranja
	Oreo
<i>Cupcakes</i>	Boston Cream
	Choco-chips
	Fresa
	Polar Bear Paw
	Red Velvet
<i>Muffins</i>	Arándano
	Avena
	Cereza
	Chocolate
	Frambuesa
<i>Donas</i>	Arcoíris
	Arequipe
	Chocolate Glaseado
	Mermelada de Mora

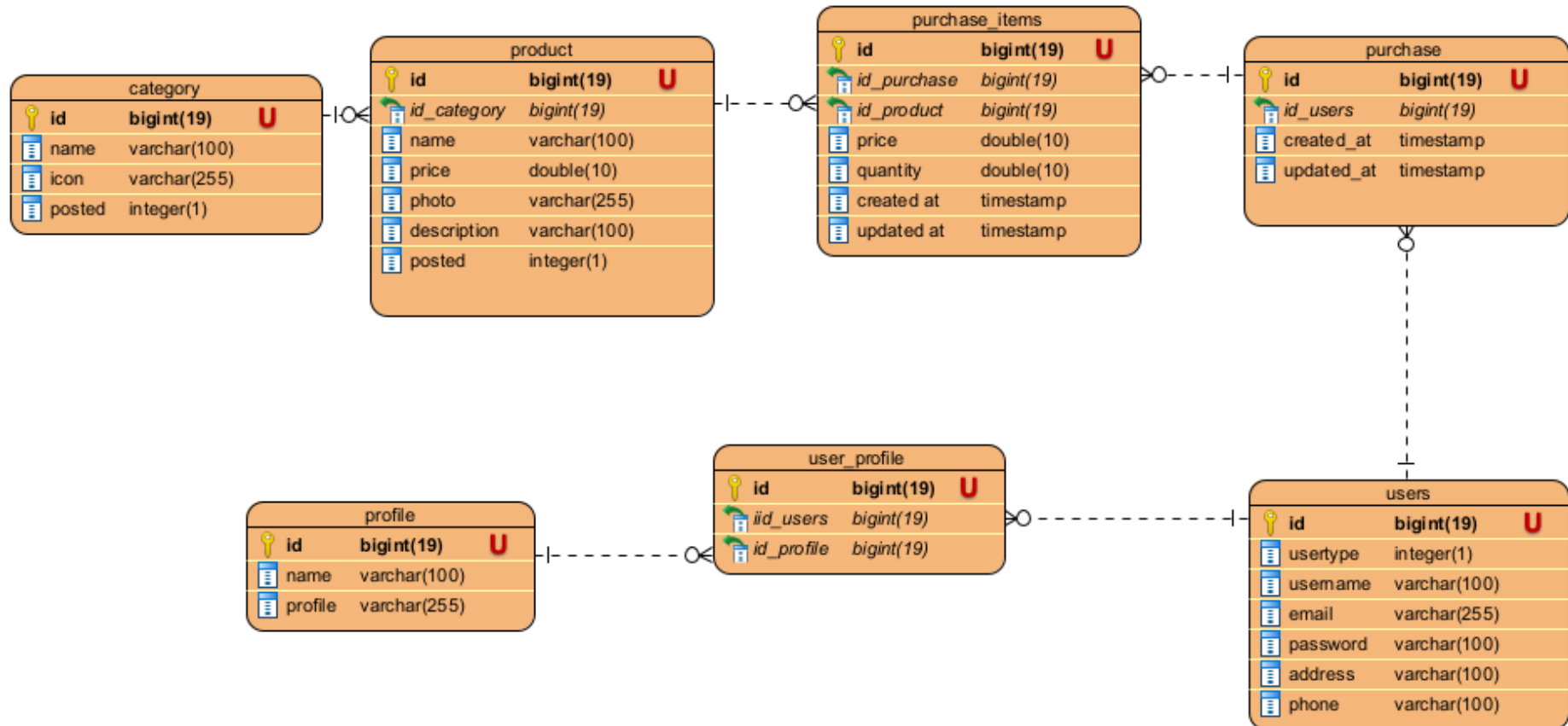
	Oreo
Otros	Alfajor
	Cinnamon roll
	Fresas con crema
	Macaron
	S'more Clásico
	S'more de Oreo

DICCIONARIO DE DATOS

Entidad	Atributo*	I D	Tipo	Tamaño	Descripción
<i>Category</i>	id	id	bigserial	8 byte	
	name		varchar	100	
	icon		varchar	255	
	posted		int	4 byte	Default = 0
<i>Product</i>	id	id	<u>bigserial</u>	8 byte	
	id_category		bigserial	8 byte	
	name		varchar	100	
	price		double precision	8 byte	
	photo		varchar	255	
	description		varchar	255	
	posted		int	4 byte	Default = 0
<i>Users</i>	id	id	bigserial	8 byte	
	usertype		int	4 byte	Default = 0
	username		varchar	100	
	email		varchar	255	
	password		varchar	100	
	address		varchar	100	
	phone		varchar	100	
<i>Purchase</i>	id	id	bigserial	8 byte	
	id_user		bigserial	8 byte	
	created at		timestap without time zone	—	Default now ()
	update at		timestap without time zone	—	Default now ()
<i>Purchase_items</i>	id	id	bigserial	8 byte	
	id_purchase		bigserial	8 byte	
	id_product		bigserial	8 byte	
	price		double precision	8 byte	
	quantity		double precision	8 byte	

**Todos los atributos solicitados SON obligatorios.*

MODELO ENTIDAD - RELACIÓN



HISTORIA DE USUARIOS

CON BASE EN LOS PRODUCTOS OFRECIDOS (entidades: category, product)

- **Admin:**

1. El usuario administrador podrá ver tanto las categorías como los productos de la tienda virtual.
2. El usuario administrador podrá filtrar los productos por categorías.
3. El usuario administrador podrá ver la descripción del producto.
4. El usuario administrador podrá ver el precio de los productos.
5. El usuario administrador podrá generar nuevas categorías, así como nuevos productos (create).
6. El usuario administrador podrá modificar/actualizar las categorías y productos existentes de la tienda virtual (update).
7. El usuario administrador podrá eliminar categorías y productos de ser necesario (delete).

- **Client:**

1. El usuario cliente podrá ver tanto las categorías como los productos de la tienda virtual.
2. El usuario cliente podrá filtrar los productos por categorías.
3. El usuario cliente podrá ver la descripción del producto.
4. El usuario cliente podrá ver el precio de los productos.
5. El usuario cliente NO podrá generar, modificar/actualizar ni borrar las categorías y productos de la tienda virtual (create, update, delete).
6. El usuario cliente podrá comprar los productos por medio del sistema de carrito de compras.

- **Guest:**

1. El usuario invitado podrá ver tanto las categorías como los productos de la tienda virtual.
2. El usuario invitado podrá filtrar los productos por categorías.
3. El usuario invitado podrá acceder a la descripción del producto.
4. El usuario invitado podrá ver el precio de los productos.
5. El usuario invitado NO podrá generar, modificar/actualizar ni borrar las categorías y productos de la tienda virtual (create, update, delete).
6. El usuario invitado NO podrá comprar los productos por medio del sistema de carrito de compras (Requiere autenticación).

CON BASE EN PERFILES Y LOGIN (entidades: users, profile, profile_users)

- **Admin:**

1. El usuario administrador deberá autenticarse en el sitio web si desea administrar la tienda.
2. El usuario administrador podrá acceder a las funciones de adición, modificación y eliminación de productos y sus categorías.
3. El usuario administrador podrá acceder a las funciones de adición, modificación y eliminación de usuarios, perfiles y perfiles de usuarios. (entidades: users, profile, users_profile).
4. La persona que posea un perfil administrador, podrá poseer también un perfil cliente en la tienda virtual si lo desea.
5. Para realizar compras en la tienda virtual, la persona con un perfil administrador deberá autenticarse con su perfil cliente en lugar de utilizar su perfil administrador.

- **Client:**

1. El usuario cliente deberá autenticarse en el sitio web si desea realizar una compra.
2. El usuario cliente podrá ver sus datos (Read).
3. El usuario cliente podrá modificar sus datos personales (username, password, email, phone, address) - Update.

- **Guest:**

El usuario invitado podrá registrarse en la tienda virtual (paso obligatorio para hacer las compras).

CON BASE EN EL CARRITO DE COMPRA (entidades: purchase, purchase_items)

- **Admin:**
 1. El usuario administrador podrá agregar los productos al carrito de compras.
 2. El usuario administrador podrá modificar la cantidad de productos que desea.
 3. El usuario administrador podrá eliminar productos del carrito de compras.
 4. El usuario administrador podrá calcular el precio total de los productos.
- **Client:**
 1. El usuario cliente podrá agregar los productos al carrito de compras.
 2. El usuario cliente podrá modificar la cantidad de productos que desea.
 3. El usuario cliente podrá eliminar productos del carrito de compras.
 4. El usuario cliente podrá calcular el precio total de los productos.
- **Guest:**

El usuario invitado NO podrá ingresar al presente servicio. Deberá registrarse si decía tener acceso a toda la funcionalidad del carrito de compras.

CONCLUSIÓN

En el proceso de realización del proyecto, se encontraron innumerables desafíos e inesperadas desgracias a enfrentar, el más grave fue la corrupción del proyecto mismo en el proceso de pulir cosas y añadir detalles, esto conllevó a tener que invertir el tiempo previsto en la recuperación parcial o total de todo lo trabajado.

Gracias a herramientas como github y a algunos backups locales, fue posible recuperar el proyecto en estado medio de desarrollo. Una vez más, las autoras del presente documento crearon los elementos y funciones perdidas. Y, finalmente, se pudo generar un resultado, si bien no el esperado ni el deseado, lo suficientemente decente para la presentación final del mismo.

Los elementos que infortunadamente no fueron implementados en el proyecto serían los siguientes: tres productos de cada categoría, carrito de compras, internalización y manejo de sesiones y cookies. Sin embargo, la tienda virtual posee una presentación atractiva e intuitiva para el usuario, es posible ver los productos en la página web y en sus secciones de categorías particulares; también es posible realizar un *login* y un *logout* con una cuenta tanto de usuario de administrador como de usuario cliente, las cuales presentan los requisitos mínimos de seguridad gracias al dominio y los mecanismos de seguridad definidos en el trabajo.

En general, la experiencia proveyó una idea de cuánta dedicación proyectos de esta índole requieren, aun si solicitudes de trabajo para realizar este tipo de cosas está normalizado y las personas del común no lo consideran gran cosa. Por otro lado, las autoras recalcan reiteradamente cuán útil y maravilloso son los backups y herramientas como github para preservar a trabajos que con tanto esmero y dedicación se le ha dedicado.

ANEXOS

PRUEBAS POSTMAN PARA EL CRUD

Adicionar una categoría - CREATE (método post)

The screenshot displays the Postman interface for a REST client. At the top, the collection is named 'cake_shop' and the endpoint is 'CREATE_CATEGORY'. The request method is 'POST' and the URL is 'http://localhost:8080/cakeshop/prueba/category ...'. The 'Body' tab is selected, showing a JSON payload:

```
{  "name": "postres",  "icon": "postres.png"}
```

. The status bar indicates a successful response with 'Status: 200 OK', 'Time: 75 ms', and 'Size: 173 B'. The response body is shown in the 'Pretty' view as:

```
{  "Mensaje": "Se adicionó correctamente."}
```

cake_shop / CREATE_CATEGORY

Save

POST http://localhost:8080/cakeshop/prueba/category ... Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "name": "postres",
3   "icon": "postres.png"
4 }
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 75 ms Size: 173 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "Mensaje": "Se adicionó correctamente."
3 }
```

Traer todas las categorías - READ (método get)

cake_shop / GET_ALL_CATEGORIES

GET ▼ http://localhost:8080/cakeshop/prueba/category Send ▼

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
-----	-------	-------------	-----	-----------

Body Cookies Headers (4) Test Results 🌐 Status: 200 OK Time: 292 ms Size: 521 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ ≡

```
1  {
2    "categories": [
3      {
4        "icon": "torta01.png",
5        "id": 1,
6        "name": "Tortas"
7      },
8      {
9        "icon": "pies.png",
10       "id": 2,
11       "name": "Pies"
12     },
13   ]
14 }
```

🔍 Capture requests and cookies 🏠 Bootcamp 🏃 Runner 🗑️ Trash 🔧

Actualizar una categoría pasando por parámetro el id - UPDATE (método put)

cake_shop / PUT_CATEGORY_ID

PUT [Send](#)

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** [Beautify](#)

```
1 {
2   ... "id": "8",
3   ... "name": "heladitos",
4   ... "icon": "helado01.png"
5 }
```

Body Cookies Headers (4) Test Results [Save Response](#)

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "Mensaje": "Se modificó correctamente."
3 }
```

Borrar una categoría pasando por parámetro el id - DELETE

cake_shop / DELETE_CATEGORY Save ... ✎ 💬

DELETE ▼ http://localhost:8080/cakeshop/prueba/category/9 Send ▼

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (4) Test Results 🌐 Status: 200 OK Time: 154 ms Size: 172 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 🔗 📄 🔍

```
1  {
2    "Mensaje": "Se eliminó correctamente."
3  }
```

