

APLICACIÓN CLIENTE SERVIDOR

Rodrigo N. Santander

Gonzalo Lastra

UNIVERSIDAD PROVINCIAL DEL SUDOESTE

FACULTAD DE LA PEQUEÑA Y MEDIANA EMPRESA

TECNICATURA EN TECNOLOGÍAS DE LA PROGRAMACIÓN

CÁTEDRA: PROYECTO INFORMÁTICO



Docente: Carlos Berger

Monte Hermoso, Buenos Aires, Argentina

Noviembre de 2024

Resumen.....	4
Introducción.....	4
Marco teórico.....	4
Lenguajes de Programación.....	4
Python.....	4
JavaScript.....	5
Paradigmas de Programación.....	5
Programación Funcional.....	5
Programación Orientada a Objetos (POO).....	5
Flask.....	5
HTML.....	6
CSS.....	6
Bases de Datos.....	6
Relacionales (SQL).....	6
SQL (Structured Query Language).....	6
No Relacionales.....	7
Entornos de Desarrollo Local.....	7
XAMPP / LARAGON.....	7
Control de Versiones.....	7
Git.....	7
El Protocolo de Hipertexto (HTTP).....	7
Métodos GET, PUT, POST, DELETE, PATCH.....	8
Interfaz de Aplicaciones Programables (API).....	8
JSON (JavaScript Object Notation).....	8
Patrones de diseño.....	9
Inyección de Dependencias.....	9
Servicios.....	9
Modelos.....	9
Controladores.....	10
Reflexión.....	10
Contenedores y Docker.....	10
Contenedores.....	10
Docker.....	11
Metodología.....	11
Conclusión.....	13
Anexo A.....	15

Modo de uso de la aplicación:.....	15
------------------------------------	----

Resumen

En el desarrollo de este trabajo se desarrolló una aplicación

Introducción

En este proyecto abordaremos el desarrollo de una aplicación cliente-servidor utilizando el framework flask para el lenguaje Python, el cual es el lenguaje principal abordado durante el desarrollo de la carrera, para la tecnología backend que será una API RESTfull, en el cual para la comunicación con el cliente utilizaremos un frontend basado en html /css/js.

Las pautas para el desarrollo es que se requiere de un login que permite acceder al sistema como administrador o cliente. El administrador tendrá la facultad de acceder a los pedidos realizados por el cliente mediante un menú que también tendrá la posibilidad de modificar.

El cliente tiene la posibilidad de revisar el menú, hacer un pedido, y pagar el mismo mediante la implementación de un carrito, una vez confirmado este puede ver los pedidos realizados y cancelarlos si no están en preparación

Marco teórico

Lenguajes de Programación

Python

Python es un lenguaje de programación de alto nivel, interpretado y con una sintaxis simple y fácil de aprender. Es muy utilizado en el desarrollo de aplicaciones web, análisis de datos y desarrollo de APIs, como en el caso del backend de este proyecto, donde se emplea el framework Flask para crear una API RESTful. Python es conocido por su flexibilidad y por ser el lenguaje principal en este desarrollo debido a su robustez y la amplia disponibilidad de bibliotecas.

JavaScript

JavaScript es un lenguaje de programación interpretado que permite la creación de contenido dinámico en la web. Es fundamental para el desarrollo del frontend, ya que permite manipular elementos HTML y CSS de manera interactiva, como en el caso de la comunicación cliente-servidor en este proyecto, donde JavaScript se encarga de enviar solicitudes al backend mediante el uso de API.

Paradigmas de Programación

Programación Funcional

La programación funcional es un paradigma que trata de minimizar los efectos secundarios y utiliza funciones puras, es decir, funciones que siempre devuelven el mismo resultado con los mismos parámetros. En proyectos web como el presente, se pueden aplicar principios de programación funcional en el desarrollo de funciones que gestionan la lógica del negocio, mejorando la mantenibilidad y la modularidad del código.

Programación Orientada a Objetos (POO)

La programación orientada a objetos es un paradigma que organiza el código en "objetos", los cuales combinan datos y comportamientos. Este enfoque es útil en el desarrollo de aplicaciones complejas como la gestión de usuarios y pedidos en este proyecto. Flask, como framework Python, permite integrar la programación orientada a objetos para estructurar el backend de manera eficiente.

Flask

Flask es un framework web ligero y flexible para Python que permite desarrollar aplicaciones web de manera rápida y sencilla. Flask sigue el enfoque micro-framework, lo que significa que incluye solo los componentes esenciales para construir una aplicación web, dejando a los

desarrolladores la libertad de agregar extensiones según las necesidades del proyecto. Esta flexibilidad hace de Flask una excelente opción para proyectos que requieren simplicidad sin sacrificar la capacidad de expansión.

HTML

HTML (HyperText Markup Language) es el estándar para la creación de páginas web. En este proyecto, HTML será utilizado para estructurar el contenido y la interfaz de usuario (UI), creando páginas donde el cliente podrá ver el menú, realizar pedidos y gestionar su cuenta.

CSS

CSS (Cascading Style Sheets) es un lenguaje de estilo utilizado para dar formato y diseño a los documentos HTML. En el frontend de este proyecto, CSS será fundamental para darle un estilo atractivo y funcional al sitio web, asegurando una experiencia de usuario (UX) agradable.

Bases de Datos

Relacionales (SQL)

Las bases de datos relacionales, como MySQL o PostgreSQL, utilizan un modelo de tablas para almacenar datos de manera estructurada. En este proyecto, una base de datos relacional podría ser utilizada para almacenar la información de usuarios, pedidos, y productos, permitiendo consultas complejas a través de SQL.

SQL (Structured Query Language)

Es un lenguaje estándar utilizado para interactuar con bases de datos relacionales. SQL permite crear, leer, actualizar y eliminar datos (operaciones CRUD) en bases de datos.

SQL se utiliza para gestionar la información de los usuarios, pedidos y pagos en una base de datos relacional. Con SQL, podemos realizar consultas complejas para obtener los datos que necesitamos, así como realizar modificaciones en la base de datos de manera eficiente.

No Relacionales

Las bases de datos no relacionales, como MongoDB, utilizan estructuras más flexibles, como documentos o colecciones. Estas bases de datos son útiles cuando se necesita escalabilidad y flexibilidad en los tipos de datos almacenados, lo cual podría ser útil si se desea escalar horizontalmente la aplicación.

Entornos de Desarrollo Local

XAMPP / LARAGON

XAMPP y Laragon son entornos de desarrollo local que permiten configurar servidores web, bases de datos y otros servicios de desarrollo, nos van a facilitar el desarrollo de aplicaciones web de manera local, simulando un entorno de producción.

Control de Versiones

Git

Git es un sistema de control de versiones distribuido utilizado para gestionar y rastrear cambios en el código fuente durante el desarrollo de software. Permite a los equipos colaborar de manera eficiente, gestionando versiones de código, ramas y cambios de forma controlada. Git es esencial para mantener un registro de las modificaciones y facilitar la colaboración entre los desarrolladores.

El Protocolo de Hipertexto (HTTP)

HTTP (HyperText Transfer Protocol) es el protocolo utilizado para la transferencia de datos en la web. Las aplicaciones cliente-servidor se comunican a través de solicitudes HTTP, donde el cliente (navegador) realiza solicitudes al servidor y reciba respuestas.

Métodos GET, PUT, POST, DELETE, PATCH

Estos son los métodos fundamentales en las solicitudes HTTP:

- **GET**: Recupera datos del servidor.
- **POST**: Envía datos al servidor para ser procesados.
- **PUT**: Actualiza recursos existentes en el servidor.
- **DELETE**: Elimina recursos del servidor.
- **PATCH**: Realiza una actualización parcial de un recurso.

Interfaz de Aplicaciones Programables (API)

Una API es un conjunto de reglas y definiciones que permite que diferentes sistemas se comuniquen entre sí. En este proyecto, se implementará una API RESTful utilizando Flask para manejar las solicitudes de los clientes, así como la gestión de pedidos y usuarios. La API permitirá la comunicación entre el frontend (HTML, CSS, JS) y el backend (Python, Flask).

Este marco teórico cubre las tecnologías y conceptos clave utilizados en el desarrollo de tu aplicación, facilitando la comprensión de cómo cada herramienta contribuye al éxito del proyecto.

JSON (JavaScript Object Notation)

JSON es un formato ligero de intercambio de datos, fácil de leer y escribir para los humanos, y fácil de parsear y generar para las máquinas. En una API RESTful, JSON es el

formato estándar para enviar y recibir datos entre el cliente (frontend) y el servidor (backend).

Patrones de diseño

Inyección de Dependencias

La inyección de dependencias es un patrón de diseño que permite a los objetos recibir las dependencias necesarias desde el exterior en lugar de crearlas directamente. Este enfoque mejora la modularidad, facilita la pruebas unitarias y permite mayor flexibilidad en el sistema. En el desarrollo de esta aplicación, la inyección de dependencias se utiliza para proporcionar los servicios, como la gestión de pedidos y usuarios, sin que los controladores o modelos tengan que crearlos y gestionarlos directamente, permitiendo una mayor separación de responsabilidades y facilitando la reutilización del código. Esto se logra mediante configuraciones específicas en el framework, que permiten gestionar las dependencias de manera eficiente.

Servicios

El patrón de servicios se utiliza para separar la lógica de negocio del resto de la aplicación. Los servicios son responsables de procesar la información y realizar operaciones, como la validación de datos o la gestión de pedidos. En nuestro proyecto, los servicios, como OrderService o PaymentService, se encargan de manejar las operaciones relacionadas con los pedidos y pagos, dejando a los controladores solo la gestión de las solicitudes y respuestas.

Modelos

El patrón modelo se refiere a la representación de las entidades de la base de datos dentro de la aplicación. Los modelos son clases que reflejan las tablas de la base de datos, y

contienen la lógica necesaria para interactuar con ella, como las consultas y actualizaciones.

En nuestro proyecto, los modelos representan entidades clave como User (usuario) y Order (pedido).

Controladores

El patrón controlador se encarga de gestionar las solicitudes y respuestas del sistema.

Los controladores reciben las solicitudes HTTP desde el cliente y, utilizando los servicios y modelos, procesan la información antes de devolver una respuesta. En una arquitectura cliente-servidor como la que estamos desarrollando, los controladores son los encargados de recibir las acciones del cliente y devolver los resultados a través de respuestas HTTP.

Reflexión

La reflexión es un patrón que permite que el código inspeccione y modifique su propia estructura en tiempo de ejecución. En Python, se utilizan funciones como getattr, setattr, hasattr y type() para lograr la reflexión. Este patrón es útil cuando se necesita una mayor flexibilidad y dinamismo en la aplicación, como la creación de formularios o validadores automáticos basados en los atributos de los modelos o las solicitudes del cliente.

Contenedores y Docker

Contenedores

Los contenedores son una tecnología de virtualización a nivel de sistema operativo que permite empaquetar una aplicación junto con todas sus dependencias, librerías y configuraciones necesarias para ejecutarse de manera aislada en cualquier entorno. Los contenedores proporcionan una forma eficiente y portátil de ejecutar aplicaciones,

garantizando que se comporten de manera consistente sin importar el entorno en el que se desplieguen.

A diferencia de las máquinas virtuales, los contenedores no requieren un sistema operativo completo, sino que comparten el núcleo del sistema operativo host. Esto hace que los contenedores sean más ligeros y rápidos de iniciar, lo que mejora la eficiencia y reduce el consumo de recursos.

Docker

Docker es una plataforma de código abierto para la automatización de la implementación, escalado y gestión de aplicaciones mediante contenedores. Docker facilita la creación, despliegue y ejecución de aplicaciones en contenedores, lo que permite que los desarrolladores puedan empaquetar sus aplicaciones junto con sus dependencias en una imagen, que luego puede ser ejecutada en cualquier máquina que tenga Docker instalado.

Metodología

Para comenzar el desarrollo de la aplicación primero se levantó un entorno de local de trabajo utilizando la suite Laragon, que nos provee las herramientas necesarias para el desarrollo como Apache, MariaDB y otras utilidades para la gestión del entorno. Se creó un virtual host en la configuración de apache para poder utilizar un dominio personalizado de testeo, se crearon los certificados para manejar la encriptación SSL y permitir el HTTPS en apache, usando la herramienta openssl en consola, luego se añadió al archivo Hosts de windows una entrada para permitir la resolución del nombre.

Luego se procedió a instalar el framework de python de Flask en un entorno virtual de python, para comenzar con la construcción del backend.

Para facilitar el trabajo en equipo y asegurar la trazabilidad de los cambios, se utilizó GIT como sistema de control de versiones. Se creó un repositorio en GitHub para centralizar el desarrollo del proyecto. De esta forma, todos los cambios realizados fueron documentados mediante commits.

El cliente web se implementó utilizando JavaScript en un modelo de SPA (Single Page Application) para reutilizar y modularizar vistas y código, también nos permite garantizar la interactividad de la interfaz. La misma fue diseñada como responsive, adaptándose al formato requerido para su utilización tanto en dispositivos móviles como en pantallas de PC. Además, se implementaron validaciones en el cliente para garantizar que los datos enviados al servidor fueran correctos y cumpliera con los requisitos del sistema antes de enviar las peticiones.

Una vez preparado el entorno de trabajo, se comenzó con el diseño de los wireframes del frontend, que serían los modelos de referencia para el desarrollo de la aplicación web. Esto nos permitió que el desarrollo del frontend sea más dinámico y con una clara visión de la estructura y funcionalidad deseada. Posteriormente, se realizaron revisiones iterativas de los wireframes con el equipo y el cliente, incorporando feedback y asegurando que el diseño cumpliera con los requisitos funcionales y estéticos antes de proceder a la implementación final. Además, se utilizó un enfoque modular para el código del frontend, lo que facilitó la colaboración y la integración continua del trabajo de distintos desarrolladores.

Para el despliegue de la aplicación en un entorno de producción, se utilizará una imagen Docker, la cual puede ser desplegada en una instancia EC2 de AWS. Esto nos permite ejecutar la aplicación de manera aislada y escalable, aprovechando la infraestructura de Amazon Web Services para garantizar la disponibilidad y el rendimiento adecuado de la aplicación en el ciberespacio.

Conclusión

Durante el desarrollo de este proyecto hemos podido construir una aplicación cliente servidor funcional, que nos permitió poner en contexto y conjunto todos los temas abordados durante el cursado de la carrera, como tecnologías de backend como Flask, tecnologías de frontend como Javascript, manejo de HTML/CSS, manipulación de datos y bases de datos relacionales con SQL, también hemos podido incorporar el uso de Git para el control de versiones y el desarrollo colaborativo

Hemos podido comprender e implementar la integración y comunicación entre las diferentes partes de la aplicación, usar diferentes paradigmas de programación e implementación de patrones de diseño

Si bien la aplicación cumple con los requisitos y propósitos, hay muchos aspectos para mejorar, ya que por falta de experiencia o falta de tiempo no pudieron implementarse correctamente o no han sido las soluciones recomendables en su realización así como tampoco se ha llegado a implementar pruebas unitarias ni de integración. En el caso del frontend una mejora clave sería mejorar la gestión de los Handlers para que solo se ejecuten en las vistas correspondientes, también mejorar el renderizado solo de la parte que es necesaria. También se debería ajustar el tamaño de las fotos para no sobrecargar la página durante la carga de la misma, así como también delegar el servicio a una red de distribución de contenidos de terceros. En la sección de menús sería útil poder incorporar subcategorías para poder categorizar mejor los platos. Mejorar la gestión de estados de pedidos para que sean más intuitivos. También sería útil poder visualizar el menú sin estar autenticado en la aplicación para facilitar a los clientes la usabilidad y mejorar el embudo de conversión de las ventas.

Durante el desarrollo se presentaron dificultades como la gestión del enrutador encargado de gestionar las solicitudes del frontend y renderizar las vistas, el manejo de solicitudes asincrónicas,

Anexo A

Modo de uso de la aplicación:

Al ingresar tenemos la pantalla principal donde se presenta el comercio del cliente ya sea usuario o administrador, y le permite acceder a todas las funciones como el menú.

Sauce Grande Gourmet

Bienvenidos a Sauce Grande Gourmet

Sabores auténticos, experiencias inolvidables.

Déjate seducir por la mejor cocina

En Sauce Grande Gourmet, nuestros chefs combinan ingredientes frescos y de temporada para crear platos únicos que te harán vivir una experiencia gastronómica sin igual.

¿Por qué elegirnos?

- 📍 Productos locales y de alta calidad: Frescura y sabor en cada bocado.
- 👨‍🍳 Chefs expertos: Recetas tradicionales con un toque moderno.
- 🏡 Ambiente acogedor: Disfruta desde la comodidad de tu hogar.
- 📲 Fácil y rápido: Todo al alcance de tus manos con nuestra app.

[Ver Menú](#)

Para continuar, por favor [inicia sesión](#) o [crea una cuenta](#).

¿Tienes alguna pregunta? Contáctanos a través de nuestro [formulario de contacto](#).

Lo que dicen nuestros clientes

"¡La mejor experiencia gastronómica que he tenido! La comida siempre llega a tiempo y con una presentación impecable."

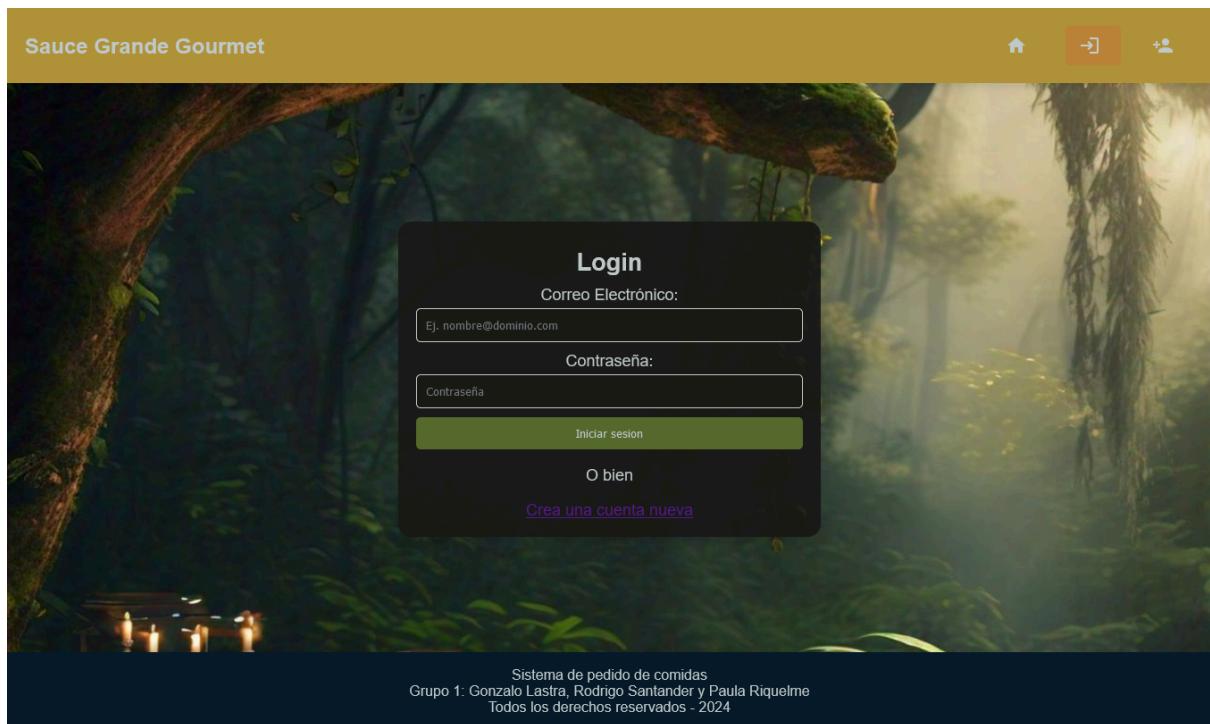
- Juan P.

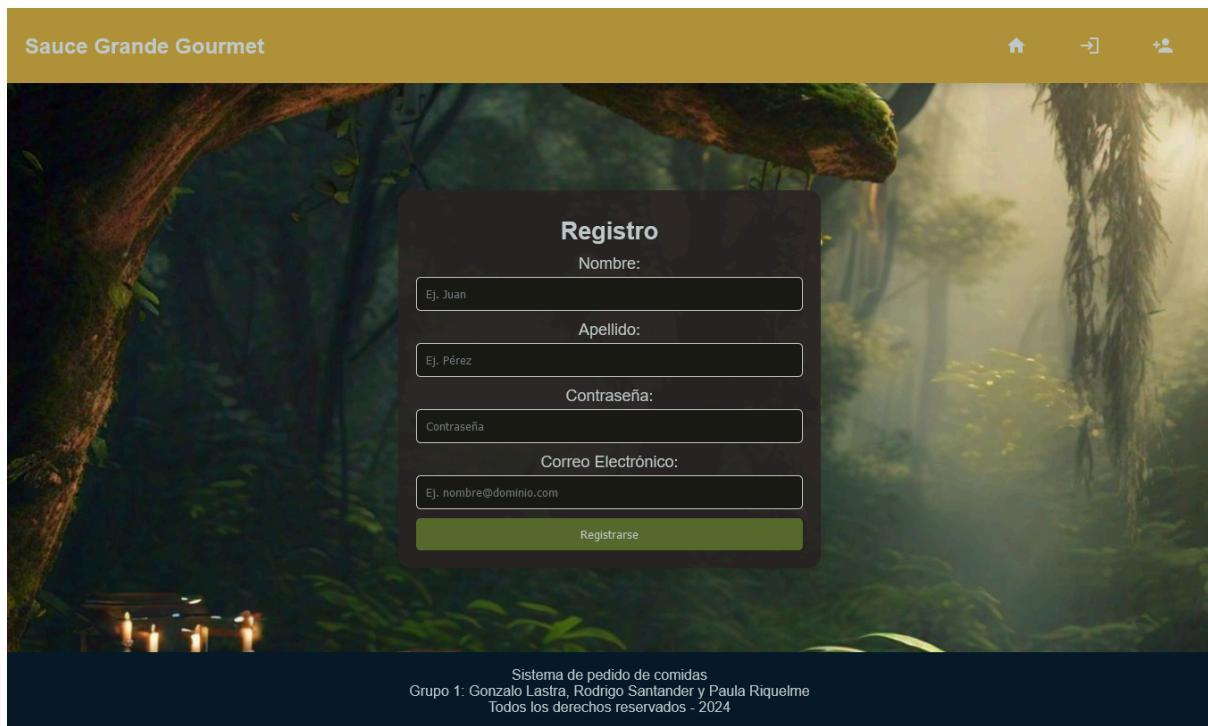
"Los platos son exquisitos y la atención al cliente es de primera. ¡Definitivamente lo recomiendo!"

- Ana G.

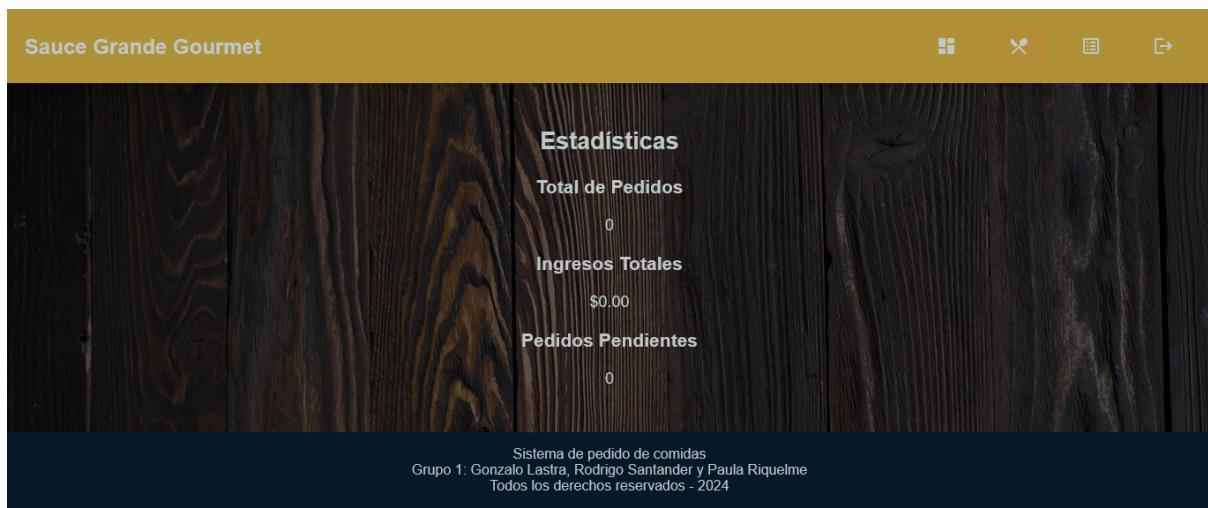
Sistema de pedido de comidas
Grupo 1: Gonzalo Lastra, Rodrigo Santander y Paula Riquelme
Todos los derechos reservados - 2024

Desde aquí puede registrarse para visualizar el menú, o loguearse si ya tiene una cuenta en el comercio. También implementamos un formulario de contacto para el público en general.





El menú Administrador le permite visualizar los pedidos realizados por el cliente, estadísticas y gestionar el menú.



Sauce Grande Gourmet

Pedidos Actuales

Pedido	Fecha y Hora	Platos	Precio Total	Estado	Acciones
Pedido #4	27/11/2024, 18:23:12	<ul style="list-style-type: none"> Cerveza artesanal, Cantidad: 1, Precio: \$350.00 Ravioles con salsa elección, Cantidad: 1, Precio: \$870.00 	\$1220.00	Creado	
Pedido #3	27/11/2024, 18:22:22	<ul style="list-style-type: none"> Patitas de pollo, Cantidad: 1, Precio: \$300.00 Lomito completo, Cantidad: 1, Precio: \$900.00 Agua mineral, Cantidad: 1, Precio: \$150.00 Vino de autor, Cantidad: 1, Precio: \$400.00 Rabas, Cantidad: 1, Precio: \$4999.00 	\$6749.00	En Progreso	
Pedido #2	27/11/2024, 18:21:14	<ul style="list-style-type: none"> Hamburguesa completa, Cantidad: 2, Precio: \$750.00 Cheesecake de frutos rojos, Cantidad: 1, Precio: \$600.00 Refresco, Cantidad: 1, Precio: \$200.00 Cerveza artesanal, Cantidad: 1, Precio: \$350.00 Volcan de chocolate, Cantidad: 1, Precio: \$5000.00 	\$7650.00	Enviado	

Pedidos Anteriores

Pedido	Fecha y Hora	Platos	Precio Total	Estado	
Pedido #1	27/11/2024, 18:19:26	<ul style="list-style-type: none"> Flan con crema y dulce de leche, Cantidad: 1, Precio: \$400.00 Ravioles con salsa elección, Cantidad: 1, Precio: \$870.00 	\$1270.00	Entregado	

Sistema de pedido de comidas
Grupo 1: Gonzalo Lastra, Rodrigo Santander y Paula Riquelme
Todos los derechos reservados - 2024

Gestión del Menú						
Añadir plato						
Imagen	Nombre	Descripción	Precio	Categoría	Acciones	
	Empanadas de carne	Empanadas rellenas de carne especiada, cocidas al horno	200.00	Entradas	Edit	Delete
	Bruschettas	Pan tostado con tomate fresco, albahaca y aceite de oliva	150.00	Entradas	Edit	Delete
	Tabla de fiambres	Selección de quesos y embutidos artesanales	500.00	Entradas	Edit	Delete
	Patitas de pollo	Croquetas de pollo empanizado, servidas con salsas	300.00	Entradas	Edit	Delete
	Milanesa de ternera	Clásica milanesa de carne, servida con guarnición	600.00	Minutas	Edit	Delete
	Papas fritas	Papas fritas caseras, crujientes y doradas	350.00	Minutas	Edit	Delete
	Hamburguesa completa	Hamburguesa con queso, lechuga, tomate y aderezos	750.00	Minutas	Edit	Delete
	Lomito completo	Sandwich de lomito con huevo, jamón, queso y vegetales	900.00	Minutas	Edit	Delete
	Helado artesanal	Helado casero con sabores a elección	250.00	Postres	Edit	Delete
	Flan con crema y dulce de leche	Postre casero acompañado de crema y dulce de leche	400.00	Postres	Edit	Delete
	Tiramisù	Postre italiano de mascarpone, café y cacao	550.00	Postres	Edit	Delete
	Cheesecake de frutos rojos	Tarta de queso con coulis de frutos rojos	600.00	Postres	Edit	Delete
	Agua mineral	Botella de agua mineral natural o con gas	150.00	Bebidas	Edit	Delete
	Refresco	Lata de refresco de cola o sabor a elección	200.00	Bebidas	Edit	Delete
	Cerveza artesanal	Pinta de cerveza artesanal, varios estilos disponibles	350.00	Bebidas	Edit	Delete
	Vino de autor	vino tinto o blanco, selección de la casa	400.00	Bebidas	Edit	Delete
	Volcan de chocolate	relleno de chocolate, acompañado con frutos finos.	5000.00	Postres	Edit	Delete
	Rabas	rabas recien pescadas del mar de MH	4999.00	Entradas	Edit	Delete
	Ravioles con salsa elección	Ravioles rellenos de trucha o verdura	870.00	Pastas	Edit	Delete

Al acceder como usuario, te permite realizar pedidos mediante un carrito, y ver las órdenes creadas

Sauce Grande Gourmet

Pedidos Actuales

Pedido	Fecha y Hora	Platos	Precio Total	Estado	Acciones
Pedido #4	27/11/2024, 18:23:12	<ul style="list-style-type: none"> Cerveza artesanal, Cantidad: 1, Precio: \$350.00 Ravioles con salsa elección, Cantidad: 1, Precio: \$870.00 	\$1220.00	Creado	
Pedido #3	27/11/2024, 18:22:22	<ul style="list-style-type: none"> Patitas de pollo, Cantidad: 1, Precio: \$300.00 Lomito completo, Cantidad: 1, Precio: \$900.00 Agua mineral, Cantidad: 1, Precio: \$150.00 Vino de autor, Cantidad: 1, Precio: \$400.00 Rabas, Cantidad: 1, Precio: \$4999.00 	\$6749.00	En Progreso	
Pedido #2	27/11/2024, 18:21:14	<ul style="list-style-type: none"> Hamburguesa completa, Cantidad: 2, Precio: \$750.00 Cheesecake de frutos rojos, Cantidad: 1, Precio: \$600.00 Refresco, Cantidad: 1, Precio: \$200.00 Cerveza artesanal, Cantidad: 1, Precio: \$350.00 Volcan de chocolate, Cantidad: 1, Precio: \$5000.00 	\$7650.00	Enviado	

Pedidos Anteriores

Pedido	Fecha y Hora	Platos	Precio Total	Estado	
Pedido #1	27/11/2024, 18:19:26	<ul style="list-style-type: none"> Flan con crema y dulce de leche, Cantidad: 1, Precio: \$400.00 Ravioles con salsa elección, Cantidad: 1, Precio: \$870.00 	\$1270.00	Entregado	

Sistema de pedido de comidas
Grupo 1: Gonzalo Lastra, Rodrigo Santander y Paula Riquelme
Todos los derechos reservados - 2024

Saucé Grande Gourmet

Restaurant

Ravióles con salsa elección

Calificación: ★★★★
No comensales

Descripción
Ravióles rellenos de ricotta o queso
Precio: \$60.00

Añadir

Bebidas

Agua mineral

Calificación: ★★★★
No comensales

Descripción
Bottles of Vero agua mineral water
Precio: \$ 50.00

Añadir

Refresco

Calificación: ★★★★
No comensales

Descripción
Lata de refresco de cola o soda elección
Precio: \$6.00

Añadir

Cerveza artesanal

Calificación: ★★★★
No comensales

Descripción
Pinta de cerveza artesanal variada
Precio: \$20.00

Añadir

Vino de autor

Calificación: ★★★★
No comensales

Descripción
Vino tinto o blanco, selección de casa
Precio: \$60.00

Añadir

Checkout de carrito

Foto	Plato	Cantidad	Precio	Subtotal	Acción
	Ravioles con salsa elección	1	870.00	870	+ -
	Flan con crema y dulce de leche	1	400.00	400	+ -

Total: \$ 1270

Checkout

Monto a Pagar: \$1270.00

Nombre Completo
APRO

Correo Electrónico
User@example

Dirección
Av San martin123

Número de Tarjeta
1234567891234567

Fecha de Expiración
12/29

CVV
123

Cancelar **Pagar**

Usar APRO como nombre para simular el pago aprobado, caso contrario dará error.

Sistema de pedido de comidas
Grupo 1: Gonzalo Lastra, Rodrigo Santander y Paula Riquelme
Todos los derechos reservados - 2024

Una vez seleccionado el menú, se redirige al carrito para proceder al checkout e implementar el pago correspondiente, aprobando así la transacción. Para evitar realizar la transacción mediante un medio externo de pago, se utilizó la palabra clave "APRO" para simular la confirmación de la misma.