

CAPSTONE

Proyecto “Sistema de Gestión para Food Truck”

Datos del documento

Histórico de Revisiones

Versión	Fecha	Descripción/cambio	Autor
1	30/08/2025	Versión inicial	RVS, GPP, VGR
2	02/11/2025	Avances del Proyecto	RVS, GPP, VGR

Información General

Escuela	Informática y Telecomunicaciones
Carrera	Ingeniería en Informática
Asignatura	Capstone
Sigla	PTY4614
Sede	Plaza Oeste
Fecha de Entrega	
Docente	Cristian Espinoza Silva
Grupo Nro.	2

Integrantes

Rut	Nombre	Correo
16.808.385-8	Gustavo Pezzini Puen	gu.pezzini@duocuc.cl
11.867.330-1	René Veloso Salazar	re.veloso@duocuc.cl
21.078.145-5	Vicente Gálvez Roldán	vic.galvez@duocuc.cl

Tabla de contenido

Datos del documento.....	2
1. Introducción.....	4
2. Descripción del Proyecto	5
3. Objetivos del Proyecto	6
3.1. Objetivo General.....	6
3.2. Objetivos Específicos.....	6
4. Requerimientos.....	7
4.1. Requerimientos Funcionales	7
4.2. Requerimientos No Funcionales	8
5. Metodología	9
6. Tecnología.....	9
7. Arquitectura	10
8. Entregables	10
9. Avances del Proyecto	10

1. Introducción

El sector de Food Trucks en Chile ha experimentado un crecimiento importante en los últimos años, consolidándose como una alternativa gastronómica popular y un modelo de negocio viable para emprendedores. Sin embargo, la mayoría de estos negocios operan con sistemas rudimentarios de gestión, dependiendo de métodos manuales o aplicaciones genéricas que no se adaptan a las particularidades operativas de la venta móvil.

La gestión eficiente de un Food Truck presenta desafíos que van más allá de un negocio tradicional: operación en espacios reducidos, conectividad intermitente, necesidad de rapidez en la atención, y cumplimiento de normativas fiscales. Estos factores hacen necesaria una solución tecnológica especializada que comprenda y se adapte a la realidad operativa de este sector.

2. Descripción del Proyecto

El proyecto consiste en el diseño, desarrollo e implementación de un sistema de gestión digital para Food Trucks mediante una Aplicación Web Progresiva (PWA), que permitirá centralizar y automatizar las principales funciones del negocio:

- **Gestión de pedidos:** toma de pedidos rápida y personalizable según productos y preferencias del cliente.
- **Impresión automática:** generación de comandas y boletas conectadas a la operación en tiempo real.
- **Modo offline:** funcionamiento completo sin conexión a internet, con sincronización automática posterior.
- **Control de caja:** apertura, cierre, ingresos y egresos con reportes detallados.
- **Historial de pedidos:** registro y consulta centralizada de todas las transacciones.
- **Integración con POS:** conexión directa con terminales de pago físico, asegurando trazabilidad y evitando digitación manual de montos.
- **Soporte multiempresa y multisucursal:** administración diferenciada por perfil de usuario y gestión centralizada de múltiples puntos de venta.

El proyecto estará orientado inicialmente a un modelo de negocio especializado en café, pero diseñado con un enfoque modular y escalable que permita su aplicación en diferentes tipos de Food Trucks (hamburgueserías, heladerías, entre otros) y se desarrollará utilizando tecnologías web modernas bajo un enfoque ágil, permitiendo avances iterativos y pruebas en ambiente real para validar su funcionamiento y escalabilidad.

3. Objetivos del Proyecto

3.1. Objetivo General

Diseñar e implementar un sistema digital integral para Food Trucks que optimice la gestión de pedidos, caja y ventas, garantizando operatividad en línea y fuera de línea, escalabilidad multi-empresa/sucursal y una experiencia de usuario eficiente y segura.

3.2. Objetivos Específicos

- Facilitar la captura rápida y precisa de pedidos con opciones de personalización según tipo de producto.
- Automatizar la generación e impresión de comandas para preparación de pedidos y boletas para clientes.
- Implementar un historial completo de pedidos con capacidad para consultas y reportes detallados.
- Garantizar la funcionalidad offline con almacenamiento local y sincronización automática de datos.
- Controlar la apertura, cierre y movimientos de caja con reportes financieros confiables.
- Integrar la plataforma con terminales de pago físicos, permitiendo transacciones automáticas y seguras.

4. Requerimientos

4.1. Requerimientos Funcionales

Grupo Requerimientos	ID Requerimiento	Descripción Requerimiento
RF-001: Gestión de Pedidos y Ventas	RF-001-A	El sistema debe proporcionar una interfaz táctil optimizada para la toma rápida de pedidos, con categorización inteligente de productos y búsqueda por texto.
	RF-001-B	Debe permitir la modificación dinámica de cantidades, eliminación de ítems y aplicación de descuentos.
	RF-001-C	El cálculo de totales debe incluir impuestos (IVA) y redondeo según normativas vigentes.
	RF-001-D	Debe soportar múltiples métodos de pago (efectivo, tarjetas, transferencias) con integración a terminales POS.
RF-002: Personalización y Configuración de Productos	RF-002-A	El sistema debe permitir la definición de modificadores dinámicos por producto (ingredientes, tamaños, extras).
	RF-002-B	Debe soportar reglas de negocio configurables (combinaciones no permitidas, productos obligatorios).
	RF-002-C	La configuración de productos debe ser centralizada pero personalizable por punto de venta.
RF-003: Gestión Documentos	RF-003-A	Generación automática de comandas de cocina con información clara de productos.
	RF-003-B	Emisión de boletas cumpliendo normativas del SII chileno.
	RF-003-C	Impresión automática en impresoras térmicas con manejo de colas y reintentos ante fallos.
RF-004: Operación Offline y Sincronización	RF-004-A	El sistema debe mantener funcionalidad sin conexión a internet, incluyendo toma de pedidos e impresión.
	RF-004-B	Sincronización automática e incremental y validación de integridad de datos.
	RF-004-C	Información del estado de conectividad.
RF-005: Control de Caja y Finanzas	RF-005-A	Apertura/cierre de caja con cuadre automático y registro de diferencias.
	RF-005-B	Seguimiento de ingresos por método de pago.

Grupo Requerimientos	ID Requerimiento	Descripción Requerimiento
RF-006: Gestión Multiusuario y Multisucursal	RF-006-A	Sistema de roles granular (Administrador, Supervisor, Cajero) con permisos específicos.
	RF-006-B	Gestión centralizada de múltiples puntos de venta con datos consolidados.
	RF-006-C	Auditoría completa de acciones por usuario.

4.2. Requerimientos No Funcionales

Grupo Requerimientos	ID Requerimiento	Descripción Requerimiento
RNF-001: Rendimiento y Escalabilidad	RNF-001-A	Tiempo de respuesta inferior a 10 segundos para operaciones críticas (toma de pedidos).
	RNF-001-B	Escalabilidad horizontal para soportar múltiples puntos de venta simultáneos.
RNF-002: Seguridad y Cumplimiento	RNF-002-A	Autenticación con cifrado end-to-end para datos sensibles.
	RNF-002-B	Cumplimiento con normativas del SII para boleta electrónica.
	RNF-002-C	Backup automático y recuperación ante desastres.
RNF-003: Usabilidad	RNF-003-A	Interfaz responsive optimizada para tablets (10-12 pulgadas) con soporte táctil.
	RNF-003-B	El sistema debe ofrecer una interfaz de usuario intuitiva, eficiente y adaptada al entorno operativo de un Food Truck.
RNF-004: Disponibilidad y Confiabilidad	RNF-004-A	Disponibilidad del 99.5% en modo online y 100% en modo offline.

5. Metodología

Se propone una metodología híbrida por fases:

1.- Fase Tradicional (Inicial – Definición y Diseño):

- Inicio del proyecto
- Levantamiento de requerimientos
- Análisis y diseño de la solución
- Planificación macro del proyecto

2.- Fase Ágil (Scrum - Desarrollo, Validación y Ajustes):

- Definición del backlog del producto (inicial y refinamiento continuo)
- Sprint Planning (cada iteración de 2-4 semanas)
- Ejecución del sprint
- Sprint Review (al final de cada iteración)
- Pruebas iterativas

3.- Fase Ágil Transversal (Tablero Kanban)

- Definición y control de tareas

6. Tecnología

Se han definido las siguientes tecnologías a utilizar para el desarrollo y funcionamiento del sistema:

Componente	Tecnología
FrontEnd	<ul style="list-style-type: none"> - React Js - Workbox + Service Workers - Dexie.js
BackEnd	Django
Base de Datos	<ul style="list-style-type: none"> - Azure SQL Server - IndexedDB (uso offline)
Control de Tareas	Azure DevOps

7. Arquitectura

La arquitectura definida es Cliente/Servidor en 3 capas (Presentación/Aplicación/Datos)

8. Entregables

Los entregables serán los siguientes:

- Documentos de planificación
- Diagramas de Arquitectura y Base de Datos
- Documentos de control y pruebas.
- Sistema Web operativo (FrontEnd/BackEnd)

9. Avances del Proyecto

Los siguientes son los avances y evidencias producidos hasta la entrega nro. 2 del 25 de octubre del 2025, hasta esa fecha el proyecto presenta un avance según la carta Gantt de un 52% que se reflejan en el desarrollo efectivo de diferentes módulos y funcionalidades.

Levantamiento y planificación macro

Se completó exitosamente la recopilación de los requerimientos funcionales y no funcionales del sistema, lo que permitió definir de forma clara y detallada las expectativas del proyecto. El backlog inicial fue ingresado en Azure DevOps, facilitando el seguimiento y la priorización de tareas. Además, se elaboró una planificación macro que guía las etapas subsiguientes del proyecto, asegurando un enfoque ordenado y focalizado desde el inicio.

Análisis y diseño de base de datos

El equipo desarrolló el análisis de entidades y el diseño de diagramas ER y UML, asegurando una visión estructurada y coherente del modelo de datos. Se efectuó la configuración inicial de Azure SQL Server, lo que permitió establecer un entorno robusto y alineado con los requerimientos del negocio.

Sprint 1: Infraestructura y Backend Core

Durante este sprint se ejecutó exitosamente la configuración de entornos de desarrollo utilizando Django y Azure. Se avanzó en la creación de los modelos de datos y en la instalación de las herramientas necesarias, tales como VS Code, Node.js y el stack de frontend (Vite, React, TailwindCSS). Se implementaron las APIs básicas para el login de usuarios y la gestión de pedidos, y se realizaron pruebas unitarias sobre estos módulos, garantizando la calidad de la base funcional del backend.

Sprint 2: Frontend Core y PWA

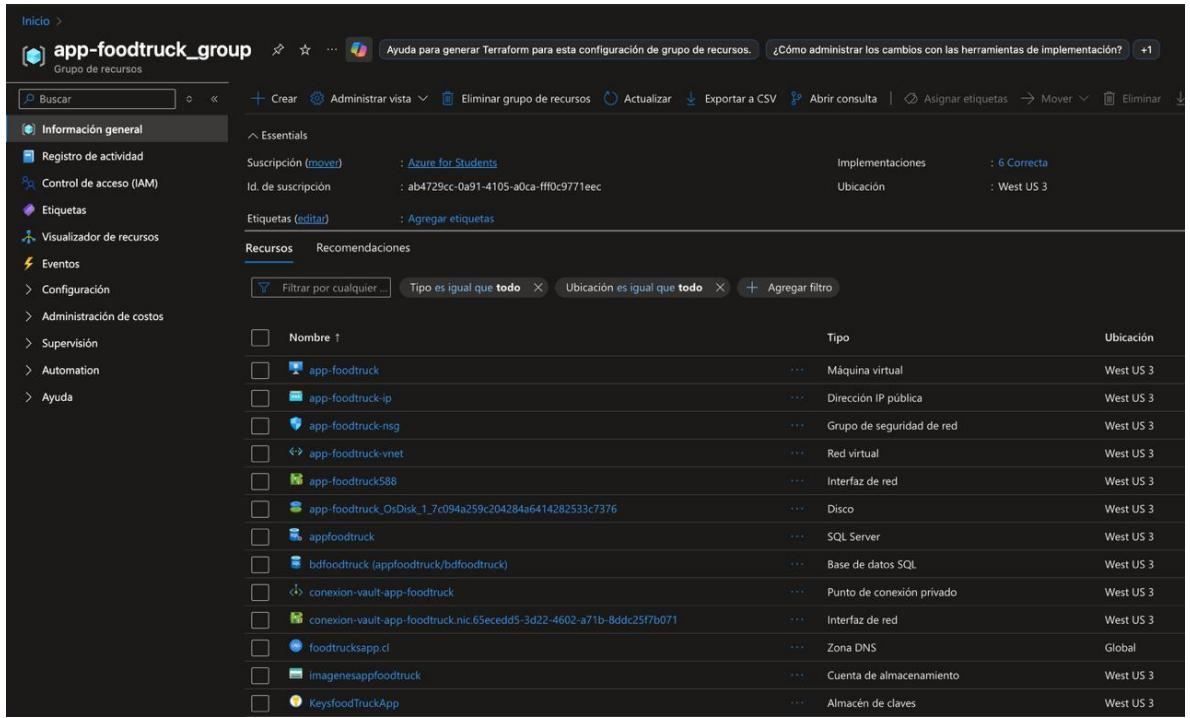
El desarrollo del frontend avanza según lo contemplado en la planificación. Ya se cuenta con la estructura base de enrutamiento y módulos clave, tales como el panel para gestión de productos y categorías. Las pantallas entregan retroalimentación clara, permiten filtrar elementos y facilitan la administración general del contenido, tal como se observa en las imágenes adjuntas.

Evidencias del Avance:

Como evidencia del progreso, se adjuntan capturas de pantalla tanto de la Carta Gantt como del grupo de recursos en Azure (app-foodtruck-group) para confirmar la implementación exitosa de la infraestructura en la región West US 3, incluyendo la máquina virtual app-foodtruck, la base de datos SQL bdfoodtruck, y las APIs (endpoints) fundamentales del sistema, todos operativos bajo la suscripción Azure for Students. Adicionalmente se incluyen imágenes del sistema en funcionamiento. Se observa que el módulo de gestión de productos ya está operativo, permitiendo el registro, edición, ocultamiento y eliminación de artículos del catálogo, así como la carga de imágenes asociadas. El módulo de gestión de categorías está implementado, permitiendo la clasificación y administración flexible de los productos. Ambas vistas permiten operar la plataforma de manera integrada y eficiente, alineándose con los objetivos establecidos en la planificación inicial.

Nombre de tarea	Duración	Trabajo	% completado	Comienzo	Fin
« Proyecto Sistema FoodTruck	84,22 días?	282 hrs	52%	lun 18/08/25	mié 17/12/25
« Levantamiento de requerimientos y planificación macro	15,88 días	63 hrs	100%	lun 18/08/25	lun 08/09/25
Recopilar requerimientos funcionales y no funcionales	27 hrs	27 hrs	100%	lun 18/08/25	jue 21/08/25
Definir backlog inicial en Azure DevOps	18 hrs	18 hrs	100%	lun 25/08/25	mié 27/08/25
Planificación macro del proyecto	23,4 hrs	18 hrs	100%	jue 28/08/25	vie 05/09/25
« Análisis y diseño de base de datos + Diagramas	8 días	60 hrs	100%	lun 01/09/25	jue 11/09/25
Análisis de entidades	24 hrs	18 hrs	100%	lun 01/09/25	lun 08/09/25
Diseño de diagramas ER y UML	18 hrs	18 hrs	100%	mié 03/09/25	vie 05/09/25
Configuración inicial de Azure SQL Server	24 hrs	24 hrs	100%	lun 08/09/25	jue 11/09/25
« SPRINT 1: Infraestructura & Backend Core	13,88 días	87 hrs	100%	lun 08/09/25	lun 29/09/25
« Configuración de Entornos y Arquitectura Inicial	4,25 días	30 hrs	100%	jue 11/09/25	mié 17/09/25
Setup de entorno Django y Azure	12 hrs	8 hrs	100%	jue 11/09/25	vie 12/09/25
Desarrollo de modelos de datos	18 hrs	18 hrs	100%	lun 15/09/25	mié 17/09/25
Instalación prerrequisitos, vs code, node.js	3 hrs	3 hrs	100%	jue 11/09/25	jue 11/09/25
Instalación vite, react, tailwindcss, dependencias	1 hr	1 hr	100%	jue 11/09/25	jue 11/09/25
Configurar vs code y extensiones	1 hr	1 hr	100%	jue 11/09/25	jue 11/09/25
Implementación de APIs básicas para Login usuarios	8 hrs	8 hrs	100%	mié 17/09/25	lun 22/09/25
CRUD gestión de pedidos	28,8 hrs	24 hrs	100%	mié 24/09/25	lun 29/09/25
Pruebas unitarias iniciales	20 hrs	24 hrs	100%	mar 30/09/25	jue 02/10/25
« SPRINT 2: Frontend Core & PWA	17,09 días	40 hrs	91%	jue 02/10/25	lun 27/10/25
Arquitectura Frontend y Enrutamiento	1,75 días	8 hrs	100%	jue 02/10/25	vie 03/10/25
Desarrollo Módulo de Autenticación	2 días	4 hrs	95%	vie 03/10/25	lun 06/10/25
Desarrollo Dashboard Vendedor y Catálogo	4 días	8 hrs	90%	vie 10/10/25	mié 15/10/25
Desarrollo Módulo Toma de Pedidos	2 días	4 hrs	90%	jue 09/10/25	jue 16/10/25
Pruebas de usabilidad	16 hrs	4 hrs	90%	vie 10/10/25	vie 24/10/25
Sprint review y refinamiento	1 día	12 hrs	70%	vie 24/10/25	lun 27/10/25

Imagen Carta Gantt hasta Entrega N° 2



The screenshot shows the Azure portal interface for the resource group 'app-foodtruck_group'. The left sidebar has a tree view with 'Información general' selected. The main area displays the 'Essentials' section with details like Subscription (mover) set to 'Azure for Students', ID of subscription 'ab4729cc-0a91-4105-a0ca-ff0c9771eec', Implementations set to '6 Correcta', and Location 'West US 3'. Below this is a table of resources:

	Nombre	Tipo	Ubicación
<input type="checkbox"/>	app-foodtruck	Máquina virtual	West US 3
<input type="checkbox"/>	app-foodtruck-ip	Dirección IP pública	West US 3
<input type="checkbox"/>	app-foodtruck-nsg	Grupo de seguridad de red	West US 3
<input type="checkbox"/>	app-foodtruck-vnet	Red virtual	West US 3
<input type="checkbox"/>	app-foodtruck588	Interfaz de red	West US 3
<input type="checkbox"/>	app-foodtruck_OsDisk_1_7c094a259c204284a6414282533c7376	Disco	West US 3
<input type="checkbox"/>	appfoodtruck	SQL Server	West US 3
<input type="checkbox"/>	bdfoodtruck (appfoodtruck/bdfoodtruck)	Base de datos SQL	West US 3
<input type="checkbox"/>	conexion-vault-app-foodtruck	Punto de conexión privado	West US 3
<input type="checkbox"/>	conexion-vault-app-foodtruck.nic.65ecedd5-3d22-4602-a71b-8ddc25f7b071	Interfaz de red	West US 3
<input type="checkbox"/>	foodtruckapp.cl	Zona DNS	Global
<input type="checkbox"/>	imagenesappfoodtruck	Cuenta de almacenamiento	West US 3
<input type="checkbox"/>	KeysfoodTruckApp	Almacén de claves	West US 3

Imagen de grupo de recursos en Azure (app-foodtruck-group)

The screenshot shows the Azure Data Studio interface for the **bdfoodtruck** database. The left sidebar contains a navigation menu with options like 'Información general', 'Registro de actividad', 'Etiquetas', 'Diagnosticar y solucionar problemas', 'Editor de consultas (versión preliminar)', 'Base de datos reflejada en Fabric (versión preliminar)', 'Visualizador de recursos', and several sections under 'Configuración' and 'Power Platform'. The main area displays a tree view of tables, starting with **dbo.Auditoria**, followed by **dbo.auth_group**, **dbo.auth_group_permissions**, **dbo.auth_permission**, **dbo.auth_user**, **dbo.auth_user_groups**, **dbo.auth_user_user_permissions**, **dbo.Boletas**, **dbo.Categorias**, **dbo.CierresCaja**, **dbo.core_empresa**, **dbo.djangoproject_log**, **dbo.djangoproject_content_type**, **dbo.djangoproject_migrations**, **dbo.djangoproject_session**, **dbo.MetodosPago**, **dbo.Modificadores**, **dbo.Pagos**, **dbo.PedidoDetalleModificadores**, **dbo.PedidoDetalles**, **dbo.Pedidos**, **dbo.ProductoModificadores**, **dbo.Productos**, **dbo.ReglasNegocios**, **dbo.Roles**, **dbo.Sucursalales**, and **dbo.Usuarios**. A message at the bottom left says 'Agregue o quite favoritos presionando Cmd+Shift+F'.

Imagen de la base de datos SQL bdfoodtruck.

```

    capstone_grupo_2 [SSH: 20.172.13.183]
models.py M views.py M auth_decorators.py M settings.py M urls.py M X
foodTruckApp > backend > settings > urls.py
13     Including another URLconf
15     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16
17     from django.contrib import admin
18     from django.urls import path
19     from core import views
20     from django.views.generic import RedirectView
21     from core.views import producto_detail, productos_list, sucursales_list, sucursal_detail, empresas_list
22
23     urlpatterns = [
24         path('api/v1/auth/login/', views.login_api, name='api_login'),
25         path('api/v1/auth/refresh/', views.refresh_api, name='api_refresh'),
26         path('api/v1/auth/logout/', views.logout_api, name='api_logout'),
27         path('api/v1/auth/me/', views.me_api, name='api_me'),
28         path('api/v1/productos/<int:producto_id>', producto_detail, name='producto_detail'),
29         path("api/v1/productos/<int:producto_id>/imagen/", producto_imagen, name="producto-imagen"),
30         path("api/v1/productos/", productos_list, name="productos-list"),
31         path('api/v1/sucursales/', sucursales_list, name='sucursales_list'),
32         path('api/v1/sucursales/<int:sucursal_id>', sucursal_detail, name='sucursal_detail'),
33         path('api/v1/empresas/', empresas_list, name='empresas_list'),
34         path('api/v1/empresas/<int:empresa_id>', empresa_detail, name='empresa_detail'),
35         path('api/v1/categorias/', categorias_list, name='categorias_list'),
36         path('api/v1/categorias/<int:categoria_id>', categoria_detail, name='categoria_detail'),
37         path('api/v1/roles/', roles_list, name='roles_list'),
38         path('api/v1/roles/<int:rol_id>', rol_detail, name='rol_detail'),
39         path('api/v1/usuarios/', usuarios_list, name='usuarios_list'),
40     ]
41

```

Imagen las APIs (endpoints) del sistema

Mostrar deshabilitados

NOMBRE	CATEGORÍA	PRECIO	TIEMPO	ACCIONES
Jugo de naranja	Jugos	4500	8 min	Editar Ocultar Eliminar
Espresso	Café	4000	3 min	Editar Ocultar Eliminar
americano	Café	2500	3 min	Editar Ocultar Eliminar
Mokaccino	Café	4000	7 min	Editar Ocultar Eliminar
capuccino	Café	3500	5 min	Editar Ocultar Eliminar

Imagen módulo de gestión de productos

The screenshot shows the 'Punto Sabor' application interface. At the top right, there are buttons for 'Supervisor' and 'Cerrar sesión'. Below the header, the page title is 'Categorías'. A sub-header 'SUCURSAL' lists 'Sucursal Centro'. There are input fields for 'Nombre' (with placeholder 'Nombre categoría') and 'Estado' (set to 'Activa'). A text area for 'Descripción' contains the placeholder 'Descripción (opcional)'. A 'Crear' button is located below these fields. The main table lists categories with columns 'ID', 'NOMBRE', 'DESCRIPCIÓN', and 'ACCIONES'. Two rows are shown: '1 Café' and '10 Jugos'. Each row has 'Editar', 'Ocultar', and 'Eliminar' buttons in the 'ACCIONES' column.

ID	NOMBRE	DESCRIPCIÓN	ACCIONES
1	Café	Café de grano, bebidas calientes en base a café.	Editar Ocultar Eliminar
10	Jugos	Jugos naturales, sin azúcar.	Editar Ocultar Eliminar

Imagen módulo de gestión de categorías

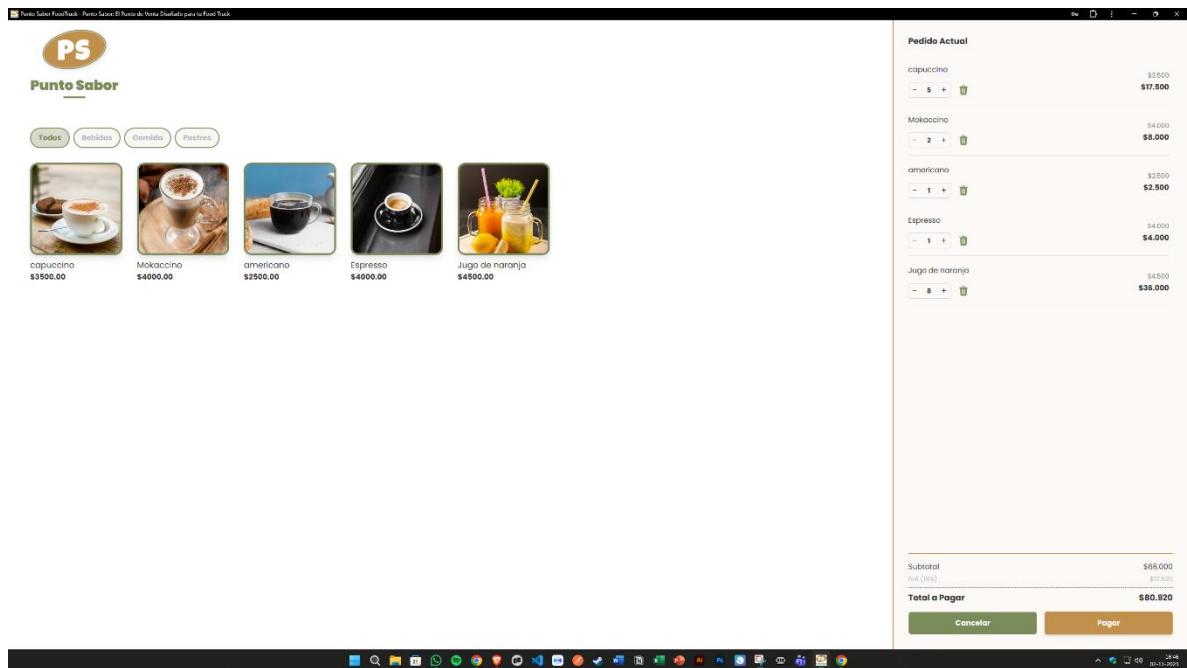


Imagen de Vista del Vendedor