



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC

**FILAVIRTUAL: SISTEMA DE GESTIÓN INTELIGENTE DE TURNOS
Y PEDIDOS PARA EL LOCAL
DE COMIDA RÁPIDA “TODO CON PAPAS”**

**Proyecto de titulación previo a la obtención del título de: Tecnólogo en
Desarrollo de Software**

Autor: Edgar Andres Caillagua Jimenez

Tutor: Jonathan David Qespaz Sanchez

Quito, Ecuador

2026

Dedicatoria

Dedico este trabajo a mi familia, que ha sido mi fortaleza y mi mayor fuente de inspiración.

A mi novia, por caminar conmigo, impulsarme y darme paz incluso en los momentos más difíciles.

Y, especialmente, a mi tía, quien me apoyó aun cuando mis decisiones no fueron las mejores. Gracias por extenderme la mano cuando más lo necesité y por recordarme que siempre puedo levantarme y seguir adelante.

Tabla de contenidos

Dedicatoria.....	2
Declaración y autorización	5
Agradecimientos	6
Resumen	7
Abstract	8
Introducción.....	10
Antecedentes.....	11
Planteamiento del Problema	12
Justificación	13
Objetivo General.....	14
Objetivos Específicos	14
Metodología.....	15
Marco Conceptual.....	17
Capítulo I	17
Capítulo II.....	20
Capítulo III.....	22
Capitulo IV	24
Conclusiones	28
Referencias bibliográficas	29
Bibliografía	29
Anexos	35

Índice de Tablas

Tabla 1: Requisitos Funcionales	30
---------------------------------------	----

Índice de Figuras

Figura 1: Proceso manual de atención al cliente en el local Todo con Papas	31
Figura 2: Arquitectura general del sistema FilaVirtual	32
Figura 3: Arquitectura general del sistema FilaVirtual	33
Figura 4: Interfaz del módulo de pedidos del sistema FilaVirtual.....	34
Figura 5: Panel de administración para gestión de pedidos.....	34
Figura 6: Interfaz del carrito de compras.....	35

Declaración y autorización

Yo, **Edgar Andres Caillagua Jimenez** con C.I. 1754790432 autor del trabajo de Integración Curricular intitulado: **“FILAVIRTUAL: SISTEMA DE GESTIÓN INTELIGENTE DE TURNOS Y PEDIDOS PARA EL LOCAL DE COMIDA RÁPIDA ‘TODO CON PAPAS’”**, previa a la obtención del título de **Tecnólogo Superior en Desarrollo de Software** en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 26 de noviembre de 2025



Edgar Andres Caillagua Jimenez

C.I. 1754790432

Agradecimientos

Quiero agradecer de corazón a mi familia, quienes han sido mi apoyo y motivación constante.

A mi madre y mi abuelita, por su cariño y dedicación que siempre me han acompañado.

A mi padre, por sus consejos y por no dejarme de lado.

A mi tía y mi tío, por su ayuda en los momentos más importantes y por estar presente en mi formación.

A mi novia, por creer en mí, por comprenderme y por acompañarme con amor y paciencia durante todo este proceso.

Y también me agradezco a mí mismo, por no rendirme y seguir avanzando a pesar de los obstáculos.

Resumen

FilaVirtual se diseñó para digitalizar el manejo de turnos y pedidos en el local Todo con Papas, eliminando el sistema manual que usan actualmente. El problema principal que encontré es que, al anotar todo a mano, se arman filas innecesarias, los clientes se amontonan y el control de lo que se vende es muy desordenado. Todo este caos no solo molesta a la gente que va a comer, sino que también confunde a los empleados, causando retrasos y haciendo que el servicio sea mucho más lento de lo que debería.

El sistema busca automatizar el proceso de atención mediante la asignación de turnos virtuales, la visualización del estado de los pedidos y el envío de notificaciones internas en tiempo real. Para su desarrollo se empleará un stack tecnológico compuesto por React.js en el frontend, Node.js con Express en el backend y PostgreSQL como gestor de base de datos. Adicionalmente, se integrará Socket.io para facilitar la comunicación instantánea entre los clientes y el personal del local.

La metodología seleccionada para el desarrollo es Scrum, organizada en doce sprints de dos semanas cada uno. Este enfoque permitirá avanzar de forma incremental, evaluar continuamente el producto y realizar ajustes de acuerdo con las necesidades del negocio. Con la implementación de esta solución se espera reducir en al menos un 30% los tiempos de espera, mejorar la organización interna del establecimiento y ofrecer a los clientes una experiencia de atención más ágil, clara y ordenada.

Palabras claves

- Gestión de turnos
- Pedidos digitales
- Sistema web responsivo
- Tiempo real
- Experiencia de usuario
- Automatización

Abstract

This project aims to develop FilaVirtual, a responsive web system designed for the digital management of customer queues and orders at the fast-food establishment Todo con Papas. Currently, the business faces several operational challenges derived from manual service processes, including long waiting times, customer congestion, and disorganized order tracking. These issues negatively affect both user experience and staff efficiency, leading to delays, confusion, and a decrease in service quality.

The proposed system seeks to automate the service workflow through virtual ticket assignment, real-time order status display, and internal notifications. The solution will be developed using a technological stack consisting of React.js for the frontend, Node.js with Express for the backend, and PostgreSQL as the relational database. In addition, Socket.io will be implemented to enable real-time communication between customers and staff.

The selected development methodology is Scrum, divided into twelve two-week sprints, allowing incremental progress, continuous evaluation, and timely adjustments based on business needs. This solution is expected to reduce waiting times by at least 30%, improve internal organization, and provide customers with a more agile,

transparent, and organized service experience. Furthermore, the system will generate relevant information to support operational decision-making within the business.

Keywords

- Queue management
- Digital orders
- Responsive web system
- Real-time communication
- User experience
- Automation

Introducción

En el sector de la comida rápida, la eficiencia no es un lujo sino una necesidad básica, ya que la velocidad y el orden determinan si un cliente regresa o no. Sin embargo, en el local Todo con Papas, la operatividad se ha visto frenada por el uso de métodos manuales. Esta falta de una herramienta tecnológica hace que en las horas de mayor afluencia, el local se vea desbordado, provocando errores en las entregas y una percepción de desorden que daña la imagen del negocio.

Por esta razón, este proyecto se enfoca en el desarrollo e implementación de FilaVirtual. No se trata simplemente de una página web estática, sino de una solución técnica integral diseñada para automatizar todo el flujo, desde que se toma el pedido hasta que se entrega en caja. Al digitalizar estos procesos, busco que el local deje atrás la confusión del registro manual y pase a un modelo de gestión moderno y transparente. Con esta plataforma, el cliente tiene la seguridad de saber exactamente cuánto falta para su pedido, mientras que la administración puede operar con datos precisos, eliminando los retrasos y mejorando la calidad del servicio al cliente.

Antecedentes

El negocio Todo con Papas ha experimentado un crecimiento notable en su clientela gracias a la calidad de sus productos, pero este éxito ha dejado en evidencia que su modelo de atención se quedó en el pasado. El local ha operado bajo un esquema netamente tradicional donde la comunicación entre el mostrador y la cocina depende de comandas escritas a mano y avisos verbales. Al revisar cómo llevan sus cuentas, me di cuenta de que no cuentan con ningún tipo de registro digital, todo se maneja en cuadernos o facturas físicos que muchas veces se pierden.

Esta falta de digitalización no solo es un problema de orden, sino de escalabilidad. El personal gasta gran parte de su tiempo tratando de descifrar la letra de los pedidos o gritando nombres para que los clientes retiren su comida, lo que funcionaba cuando tenían diez clientes al día, pero que ahora resulta insuficiente. No existe un antecedente de software ni de bases de datos en el local, lo que convierte a FilaVirtual en la primera intervención tecnológica real para transformar un negocio familiar en una operación mucho más profesional y eficiente.

Planteamiento del Problema

El problema en Todo con Papas aparece cada vez que llega la hora pico. Al no existir un sistema que organice la entrada de pedidos, se genera un caos logístico donde el cliente no sabe cuánto tiempo realmente va a esperar y el personal de cocina se siente presionado por el amontonamiento de gente frente al mostrador. He identificado que este desorden manual provoca errores críticos, desde entregar un combo equivocado hasta olvidar ingredientes especiales que el cliente pidió, simplemente porque la comanda de papel se mojó o se perdió en el ajetreo.

A esto se le suma la frustración del usuario. En un entorno de comida rápida, el cliente valora el tiempo, y estar parado sin saber si su turno es el siguiente o si faltan diez personas genera una mala experiencia.

Desde el lado administrativo, la situación es igual de grave: al final del día, el dueño no tiene forma de saber con exactitud cuál fue el producto más vendido o cuánta materia prima se gastó, ya que no hay un sistema que centralice la información. En definitiva, el problema es que el local está intentando manejar una demanda moderna con herramientas obsoletas, lo que frena sus ganancias y agota a su equipo de trabajo.

Justificación

La implementación de FilaVirtual es necesaria para eliminar el desorden operativo en Todo con Papas. Actualmente el uso de comandas físicas genera errores en los pedidos y aglomeraciones que afectan la experiencia del cliente. Al automatizar estos procesos, el local no solo optimiza el tiempo de entrega, sino que también profesionaliza su imagen y reduce el estrés del personal.

Desde el lado técnico, el sistema permite centralizar la información en una base de datos segura, ofreciendo al dueño un control real sobre sus ventas y stock que antes no tenía. Este proyecto justifica su desarrollo porque transforma una gestión manual ineficiente en un modelo digital moderno, escalable y capaz de mejorar la rentabilidad del negocio.

Objetivo General

Desarrollar un sistema web que ayude a organizar los turnos y pedidos del local de comida rápida Todo con Papas, con el fin de mejorar la atención al cliente, reducir los tiempos de espera y facilitar el trabajo del personal.

Objetivos Específicos

- Analizar los procesos actuales de atención, gestión de turnos y pedidos en el local Todo con Papas, identificando sus principales deficiencias.
- Diseñar la arquitectura del sistema FilaVirtual, considerando una solución web responsiva y escalable.
- Desarrollar un módulo de gestión de turnos que permita asignar y controlar el orden de atención de los clientes de manera automática.
- Implementar un módulo de gestión de pedidos que facilite el registro, seguimiento y control de los pedidos realizados.
- Integrar un sistema de notificaciones en tiempo real que permita informar a clientes y personal sobre el estado de los turnos y pedidos.

Metodología

Enfoque metodológico del proyecto

El presente proyecto se desarrolla bajo un enfoque aplicado, orientado a la solución de un problema técnico-tecnológico previamente identificado en el local de comida rápida Todo con Papas. Este enfoque permite analizar de manera concreta la situación actual del establecimiento y diseñar una solución de software que responda directamente a las necesidades detectadas en los procesos de gestión de turnos y pedidos.

Para el levantamiento de información se empleó la observación directa del proceso de atención, así como entrevistas informales con el personal del local, con el objetivo de identificar los principales inconvenientes operativos, los requerimientos funcionales del sistema y las expectativas de los usuarios finales. Esta información sirvió como base para el diseño e implementación de la solución tecnológica propuesta.

Metodología de desarrollo de software Scrum

Para el desarrollo del sistema propuesto se empleó la metodología ágil Scrum, ampliamente utilizada en proyectos de software debido a su flexibilidad, enfoque iterativo y capacidad para entregar valor de manera progresiva (Schwaber & Sutherland, 2020).

Cada sprint tuvo una duración de dos semanas y se ejecutó un total de doce sprints para completar el desarrollo del sistema. En cada ciclo se llevarán a cabo las ceremonias propias de Scrum, que son:

Planificación del Sprint (Sprint Planning): Selección de las funcionalidades a desarrollar a partir del Product Backlog.

Reuniones diarias (Daily Scrum): Revisión del avance, identificación de obstáculos y organización de las tareas diarias.

Revisión del Sprint (Sprint Review): Presentación del avance funcional del sistema al propietario del local o a los interesados.

Retrospectiva del Sprint (Sprint Retrospective): Análisis del proceso de trabajo y definición de mejoras para el siguiente sprint.

Los roles definidos dentro del marco Scrum para este proyecto son los siguientes:

Product Owner: Propietario del local Todo con Papas, responsable de definir y priorizar las funcionalidades del sistema según las necesidades del negocio.

Scrum Master: Estudiante desarrollador, encargado de facilitar la metodología, eliminar impedimentos y asegurar un flujo de trabajo eficiente.

Development Team: Conformado por el estudiante desarrollador, responsable del diseño, programación, integración y pruebas del sistema web.

Marco Conceptual

El presente marco conceptual reúne los conceptos, fundamentos tecnológicos y metodológicos que sustentan el desarrollo del sistema FilaVirtual para el local de comida rápida Todo con Papas. En este apartado se describen los elementos conceptuales aplicados al contexto del proyecto, permitiendo comprender la problemática identificada, las tecnologías empleadas, la arquitectura del sistema y el enfoque metodológico utilizado durante su desarrollo.

El marco conceptual se estructura en cuatro capítulos, los cuales abordan la situación actual de la gestión de turnos y atención al cliente, los fundamentos tecnológicos del sistema, la arquitectura y componentes desarrollados, así como la metodología aplicada, integrando de manera coherente los aspectos conceptuales y prácticos del proyecto.

Capítulo I

Situación actual de la gestión de turnos y atención al cliente en el local Todo con Papas

Descripción del local Todo con Papas

El local Todo con Papas es un establecimiento de comida rápida especializado en la venta de papas fritas y productos derivados. Se encuentra ubicado en una zona de alta afluencia de estudiantes y trabajadores, lo que genera una demanda constante de clientes a lo largo del día. El negocio se caracteriza por ofrecer un menú variado, un servicio rápido y una atención cercana al cliente.

Actualmente, el local cuenta con un personal reducido conformado por cajeros, cocineros y personal de atención al cliente, quienes gestionan de manera simultánea

pedidos presenciales y telefónicos. Esta dinámica de trabajo exige una correcta organización para evitar retrasos, errores en los pedidos y confusión durante la atención.

El horario de atención se extiende desde las 10:00 hasta las 22:00, lo que implica un flujo continuo de clientes, especialmente en horarios pico como el mediodía y la tarde. Si bien la estructura del local está pensada para una atención rápida, la ausencia de un sistema automatizado de gestión de turnos genera desorden operativo, retrasos en la atención y una experiencia poco clara para el cliente.

Proceso actual de atención al cliente

En la situación actual, la atención al cliente se realiza de manera manual mediante un sistema de fichas o turnos escritos a mano. El proceso de atención sigue los siguientes pasos: el cliente llega al local y solicita su turno en la caja; el cajero registra manualmente el nombre del cliente y la hora de llegada en un soporte físico; los pedidos se preparan según el orden de llegada, mientras los clientes esperan en una fila física hasta ser llamados para retirar su pedido; finalmente, cualquier cambio, cancelación o corrección en el pedido se gestiona de forma manual.

Este procedimiento, aunque funcional, es propenso a errores humanos, especialmente durante las horas de mayor afluencia. La falta de un registro digital limita la capacidad del personal para controlar los tiempos de espera, priorizar pedidos y mantener un orden adecuado, lo que genera insatisfacción en los clientes y sobrecarga al personal.

El flujo manual de atención se representa en la **Figura 1**, donde se pueden observar los puntos del proceso que generan demoras, confusión y desorganización durante la atención al cliente.

Problemas identificados en la gestión de turnos y pedidos

Al revisar de cerca cómo trabaja el local, me di cuenta de que el sistema manual es el principal freno para el negocio. El problema más evidente ocurre en las horas de más movimiento, donde los tiempos de espera se vuelven eternos porque no hay un orden claro. Al anotar todo en papel, siempre existe el riesgo de que las comandas se pierdan, se mojen o se dupliquen, lo que termina en reclamos de los clientes. Además, ni el dueño ni los empleados saben con certeza cuánta gente falta por atender o en qué estado real está cada pedido, lo que genera una sensación de improvisación constante.

El uso de notas a mano también pasa factura en la cocina. Es muy fácil leer mal una instrucción o confundir un pedido cuando hay presión, y eso daña directamente la calidad de lo que se entrega. Estos fallos no solo molestan al cliente, sino que hacen que el personal trabaje el doble y que el local no pueda vender más de lo que vende ahora por culpa del desorden.

Necesidad de una solución tecnológica

Los problemas identificados evidencian que ya no puede seguir dependiendo de un cuaderno y un esfero. Se necesita una herramienta que tome el control y automatice el flujo desde que el cliente llega. Un sistema digital no solo sirve para dar números, sino para que la asignación de turnos sea automática y el personal sepa exactamente qué preparar sin errores.

Al eliminar los papeles, la información fluye en tiempo real. Esto cambia totalmente la experiencia: el cliente está más tranquilo porque sabe que su turno está registrado y el personal puede enfocarse en cocinar bien en lugar de estar descifrando comandas. Implementar esta tecnología es el paso lógico para que el local sea más eficiente, se vea más profesional y deje de perder dinero por fallas operativas.

Capítulo II

Fundamentos conceptuales y tecnológicos del sistema FilaVirtual

El desarrollo de FilaVirtual no se trata solo de programar una web, sino de aplicar la transformación digital para resolver un problema real de desorden en un negocio de comida. En el mundo de la gastronomía actual, ya no basta con cocinar bien, es necesario que la información fluya rápido. Por eso me enfoqué en reemplazar los cuadernos y las notas a mano por un sistema que centralice todo y reduzca el margen de error humano, especialmente cuando el local está lleno y la presión es alta

Sistemas de gestión digital de turnos y pedidos

Un sistema de gestión digital bien hecho debe ser la base del local. En FilaVirtual, cada pedido está amarrado a un turno específico desde el segundo uno. Esto es clave porque mejora la comunicación entre la caja y la cocina: el cocinero ya no tiene que adivinar qué preparar, y el cliente tiene la seguridad de saber exactamente en qué posición de la fila está, eliminando esa incertidumbre de no saber cuánto más le toca esperar.

Experiencia de usuario y diseño responsivo

Para que el dueño y los clientes usen la app sin problemas, la interfaz tiene que ser clara e intuitiva; si es difícil de usar, nadie la va a querer. Por eso trabajé en un diseño responsivo. Esto significa que el sistema se adapta igual de bien a una computadora, a una tablet o al celular del cliente que quiere revisar su turno desde su mesa.

En cuanto a la estructura técnica, elegí una arquitectura cliente-servidor. Separé el frontend (lo que el usuario ve) y el backend (donde ocurre la lógica y manejo de datos). Esta separación es fundamental porque permite que el sistema sea estable y si en

el futuro queremos agregarle nuevas funciones, no tengamos que tumbar toda la aplicación para hacerlo.

Arquitectura del sistema y servicios web

Para el desarrollo de FilaVirtual decidí estructurar el sistema bajo una arquitectura cliente-servidor, separando claramente las funciones del frontend y del backend. El frontend actúa como la interfaz interactiva para el usuario, mientras que el backend se encarga de procesar la lógica de negocio y administrar el almacenamiento de los datos. Esta conexión se logra mediante servicios web y APIs REST bajo el protocolo HTTP, lo que me permite tener un sistema mucho más organizado, fácil de mantener y con la capacidad de escalar a futuro sin comprometer la estructura base.

Comunicación en tiempo real en aplicaciones web

Lograr que la información se actualice al instante es fundamental en un sistema de gestión de pedidos. Por esta razón integré tecnologías como WebSockets, que permiten una comunicación directa y constante entre el servidor y el cliente sin tener que estar refrescando la página manualmente. En FilaVirtual uso esta capacidad para que los cambios en el estado de los pedidos se vean reflejados de inmediato, mejorando la coordinación del personal en cocina y dándole al cliente datos reales sobre su tiempo de espera.

Bases de datos relacionales y gestión de la información

La gestión de los datos permite registrar pedidos y tiempos de atención de forma segura y consistente. Para FilaVirtual, elegí PostgreSQL como motor de base de datos debido a su gran estabilidad y a la seguridad que ofrece al manejar múltiples transacciones al mismo tiempo. Esto me garantiza que la información siempre esté íntegra y disponible. El diseño lógico que armé para estas tablas y sus relaciones se puede revisar a detalle en el Anexo B del documento.

Capítulo III

Arquitectura y componentes del sistema FilaVirtual

Para que FilaVirtual sea una herramienta estable y fácil de escalar, decidí diseñarla bajo una arquitectura cliente-servidor. Esta estructura es la que permite que la comunicación entre los clientes del local y la plataforma sea fluida, logrando que los procesos internos desde que se toma el pedido hasta que se registra en la base de datos estén bien organizados y no generen conflictos técnicos.

1. Diseño del Front-End

En la parte visual utilicé React como biblioteca principal de JavaScript. Elegí React porque trabajar con componentes reutilizables me permitió organizar el código de manera mucho más limpia y eficiente; esto es clave para hacer actualizaciones rápidas sin romper otras partes de la interfaz. Antes de escribir una sola línea de código, usé Figma para prototipar todas las pantallas, lo que me ayudó a definir cómo se vería la app y ajustar la experiencia del usuario antes del desarrollo final.

Los detalles de estas interfaces y el flujo de navegación se encuentran detallados en el **Anexo A**, **Anexo E** y en la **Figura 3**.

2. Desarrollo del Back-End

El núcleo del sistema corre sobre Node.js y Express. Estas tecnologías me permitieron construir un servidor capaz de gestionar múltiples pedidos al mismo tiempo mediante una estructura de rutas y controladores. Gracias al uso de middlewares, pude separar la lógica de negocio de la conexión con la base de datos, logrando que el procesamiento de cada solicitud sea rápido y ordenado.

Toda la configuración técnica del servidor, los endpoints y el código que hace funcionar los turnos pueden revisarse en el Anexo C, Anexo D y la Figura 2.

3. Base de datos y modelo lógico

Para el almacenamiento de la información, diseñé una estructura en PostgreSQL centrada en la trazabilidad de los pedidos. La tabla principal, llamada “orders”, es la encargada de guardar datos críticos como el código único, el monto total, el estado actual y el tiempo de espera. Por otro lado, creé la tabla “order_items” para desglosar cada producto solicitado. Este modelo relacional es lo que permite que el proceso de pago en caja y el seguimiento en pantalla funcionen sin errores.

El esquema completo y las relaciones entre tablas se detallan en el Anexo B.

4. Comunicación en tiempo real

Para la actualización en tiempo real del estado de los pedidos se integró Socket.io. Cuando el personal administrativo actualiza un pedido, el servidor emite eventos dirigidos al cliente correspondiente, permitiendo que la información se refleje de manera inmediata en la interfaz.

La lógica implementada para esta comunicación en tiempo real se evidencia en el Anexo D, donde se muestran fragmentos del código encargados de la emisión de eventos y notificaciones.

Capítulo IV

Metodología aplicada al desarrollo del sistema

El desarrollo del sistema FilaVirtual se basó en principios de metodologías ágiles, las cuales priorizan la colaboración constante, la capacidad de adaptación y la mejora continua a lo largo del proyecto.

Este enfoque permitió organizar el trabajo de forma iterativa, es decir, avanzando por etapas y realizando ajustes conforme se identificaban nuevas necesidades. De esta manera, fue posible responder de forma más eficiente a los requerimientos detectados durante el desarrollo.

Las metodologías ágiles fomentan la comunicación permanente entre los miembros del equipo y las personas involucradas en el proyecto, lo que facilita la detección temprana de posibles problemas técnicos o funcionales. Además, permiten entregar avances progresivos del sistema, validando cada funcionalidad con base en las necesidades reales del entorno del local.

La aplicación de estas prácticas ayudó a estructurar mejor el proceso de desarrollo, definir responsabilidades y mantener un control constante sobre el avance del proyecto. La evolución gradual del sistema y las pruebas realizadas sobre sus principales funcionalidades pueden observarse en los **Anexos A, C y D**, donde se evidencian los resultados obtenidos.

En general, este enfoque metodológico permitió construir una solución alineada con las necesidades del local Todo con Papas, logrando un sistema funcional, organizado y con posibilidad de seguir mejorando en el futuro.

Diseño funcional del sistema

El diseño funcional de FilaVirtual se centró totalmente en la experiencia del usuario, buscando que el proceso de compra en Todo con Papas sea intuitivo y no genere dudas. Me enfoqué en crear un flujo que acompañe al cliente desde que mira el menú hasta que recibe su comida. El proceso es sencillo: el usuario entra al catálogo, elige sus combos y confirma el pedido; en ese instante, el sistema le asigna un código único que sirve para pagar en caja y rastrear su turno. Para ver cómo quedó la interfaz final desde el lado del cliente, se pueden revisar las capturas del Anexo A.

Por otra parte, diseñé un panel administrativo para que el personal del local tenga el control total. Desde aquí pueden ver qué pedidos están entrando, cambiarles el estado o ajustar el tiempo de entrega si la cocina está muy llena. Esta herramienta es la que realmente elimina los gritos y la confusión entre el mostrador y los cocineros.

Flujo de operación del sistema

Módulo Cliente (Menú, Carrito, Mi Pedido)

Para que **FilaVirtual** mantenga el orden, establecí un flujo lógico de cinco pasos que asegura que nada se pierda:

1. El cliente navega por el menú digital y arma su carrito.
2. El sistema procesa la selección y genera el código de identificación del pedido.
3. Con ese código, el cliente se acerca a la caja para cancelar su consumo.

4. El administrador activa el pedido en el panel y actualiza el avance de la preparación.
5. El cliente monitorea en su propia pantalla cómo progresa su orden hasta que está lista.

Este flujo garantiza que el proceso sea transparente y que el local no se llene de gente preguntando cuánto falta, ya que la información está disponible en tiempo real.

Seguridad y control de acceso

Aunque es un sistema para un local comercial, no descuidé la seguridad.

Implementé un sistema de autenticación para que solo el personal autorizado pueda entrar al panel de administración. Esto evita que cualquier persona ajena al negocio pueda manipular los pedidos o borrar información. Además, puse filtros y validaciones en los formularios para que los datos que entren sean correctos y no causen errores en la base de datos.

Los detalles técnicos de cómo protegí estos endpoints están en el **Anexo C**.

Pruebas y validación del sistema

Para estar seguro de que el sistema no fallara en plena hora pico, realicé varias pruebas funcionales. Me aseguré de testear el registro de pedidos, que el código de búsqueda funcionara siempre y que la sincronización en tiempo real no tuviera retrasos. Usé herramientas como Postman para estresar el backend y verificar que los datos que enviaba el servidor fueran los correctos antes de mostrarlos en la web.

Los resultados y las evidencias de que el sistema responde bien están en el **Anexo F**. También hice pruebas de diseño responsivo, confirmando que la aplicación se

ve y funciona perfecto tanto si el cliente la abre desde un celular moderno como si el administrador la usa en una computadora de escritorio.

Evidencias del desarrollo

Para respaldar todo el trabajo técnico realizado en **FilaVirtual**, he organizado una serie de anexos que documentan cada etapa del sistema. Estas evidencias no son solo capturas de pantalla, sino la prueba real de que los objetivos planteados se cumplieron paso a paso. La documentación se divide de la siguiente manera:

Anexo A: Muestra toda la interfaz del frontend y cómo navega el usuario.

Anexo B: Detalla la estructura lógica y relacional de la base de datos.

Anexo C: Expone el código del backend y la arquitectura lógica.

Anexo D: Documenta las pruebas de funcionamiento y la mensajería en tiempo real.

Anexo F: Contiene los registros de las pruebas de endpoints realizadas en Postman.

Conclusiones

Al implementar esta solución web responsiva, logré que el flujo de atención sea mucho más claro, eliminando la incertidumbre del cliente, ya que ahora siempre sabe qué pasa con su pedido.

Desde el punto de vista técnico, la combinación de **React**, **Node.js** y **PostgreSQL** fue un acierto total. Esta arquitectura me permitió separar la interfaz de la lógica pesada, creando un sistema modular y fácil de mantener. Además, la integración de **Socket.io** resultó ser el diferencial del proyecto, ya que permite que el cliente vea los cambios de estado y tiempos de entrega en tiempo real sin tener que estar refrescando la web.

Por otro lado, el uso de un código único para cada pedido simplificó enormemente el proceso de pago en caja y permitió que el personal localice las órdenes al instante. Para proteger la administración, el uso de JWT garantiza que solo el personal autorizado maneje estados y tiempos de entrega. El flujo de elegir, confirmar, pagar y seguir; resultó ser una experiencia simple y efectiva para el usuario.

Finalmente, este proyecto deja una base sólida para el futuro del local. Como próximos pasos, se plantea integrar pasarelas de pago digital, generar reportes automáticos de ventas y analizar métricas que ayuden al dueño a tomar mejores decisiones para su negocio.

Referencias bibliográficas

Bibliografía

- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*.
- React. (s. f.). *React Documentation*.
- Node.js. (s. f.). *Node.js Documentation*.
- Express. (s. f.). *Express Documentation*.
- PostgreSQL Global Development Group. (s. f.). *PostgreSQL Documentation*.
- Socket.io. (s. f.). *Socket.io Documentation*.
- Bootstrap. (s. f.). *Bootstrap Documentation*.
- RFC 6455. (2011). *The WebSocket Protocol*.

Lista de tablas

Código	Requisito Funcional
RF-01	El sistema debe permitir al cliente visualizar el menú de productos disponibles.
RF-02	El sistema debe permitir agregar productos al carrito de compras.
RF-03	El sistema debe generar un pedido con un código único al confirmar el carrito.
RF-04	El sistema debe registrar el pedido con estado inicial “pendiente”.

RF-05	El sistema debe permitir al cliente consultar el estado de su pedido mediante el código generado.
RF-06	El sistema debe mostrar el tiempo estimado de preparación del pedido.
RF-07	El sistema debe actualizar el estado del pedido en tiempo real.
RF-08	El sistema debe permitir al administrador visualizar la lista de pedidos en orden cronológico.
RF-09	El sistema debe permitir al administrador modificar el estado y tiempo estimado de un pedido.
RF-10	El sistema debe notificar automáticamente al cliente cuando el estado del pedido cambie.

Tabla 1: Requisitos Funcionales

Lista de figuras



Figura 1: Proceso manual de atención al cliente en el local Todo con Papas

Nota: El proceso actual de atención se realiza de forma manual, lo que genera tiempos de espera prolongados y desorganización durante horas de alta afluencia.

Backend

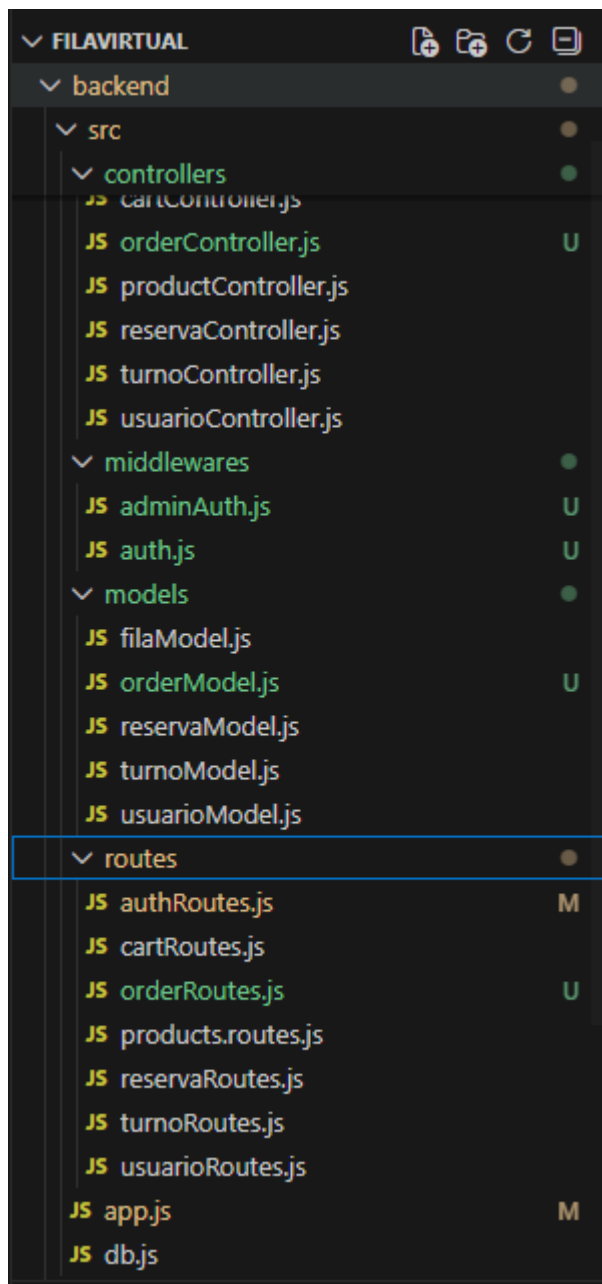


Figura 2: Arquitectura general del sistema FilaVirtual

Frontend

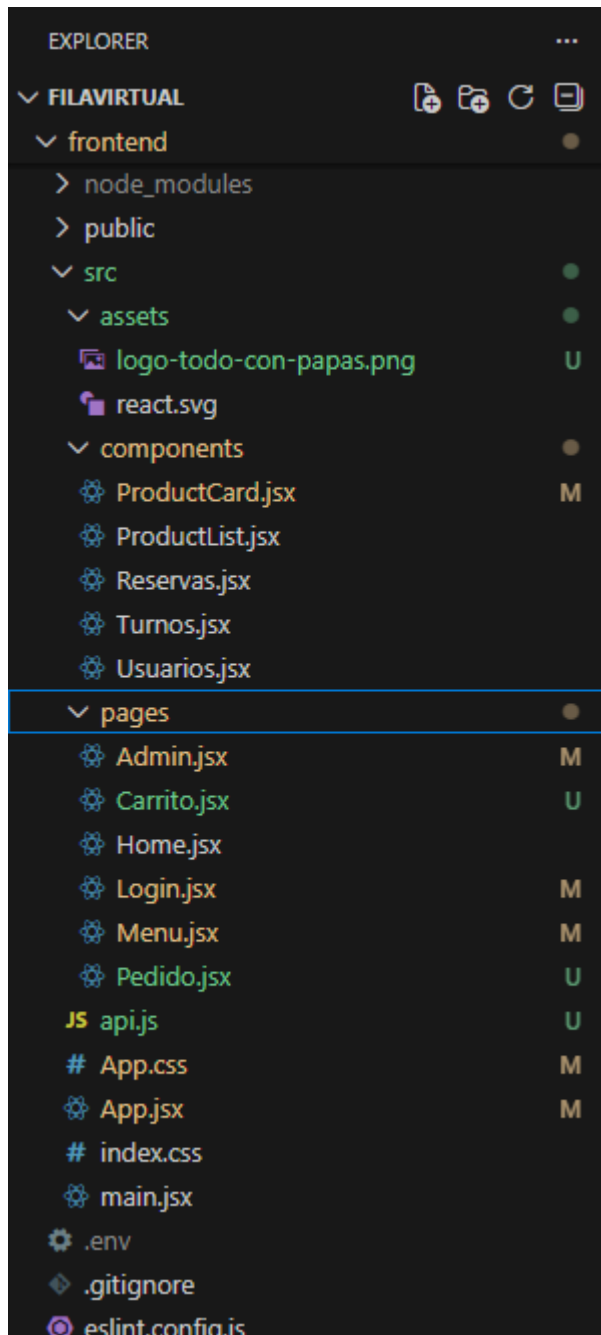


Figura 3: Arquitectura general del sistema Filavirtual

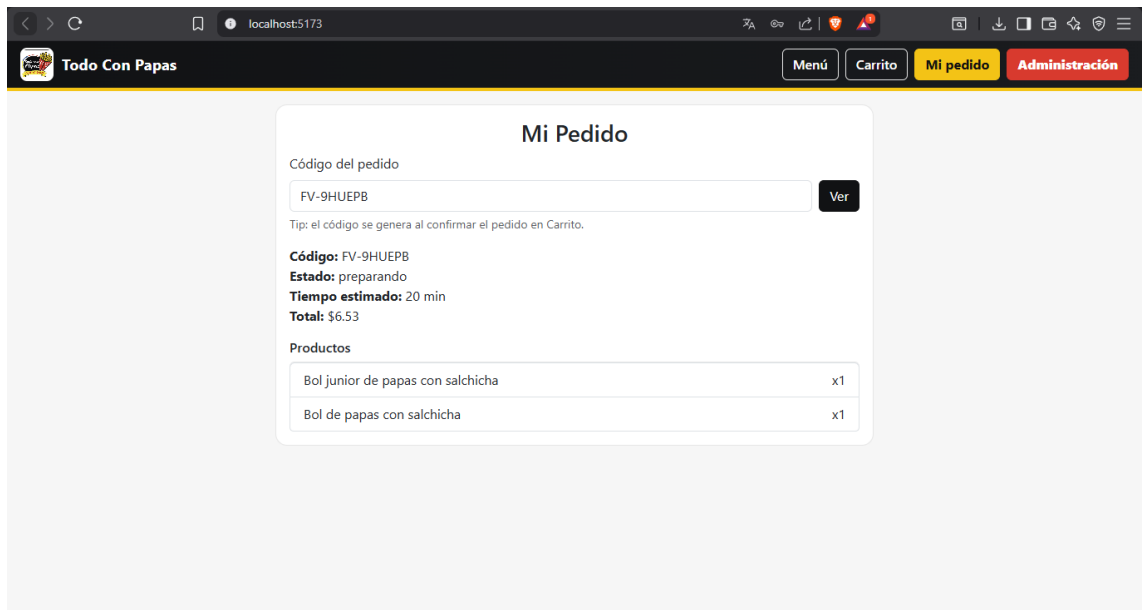


Figura 4: Interfaz del módulo de pedidos del sistema FilaVirtual

La figura presenta la interfaz del módulo de pedidos, donde el cliente puede consultar el estado y tiempo estimado de su pedido en tiempo real.

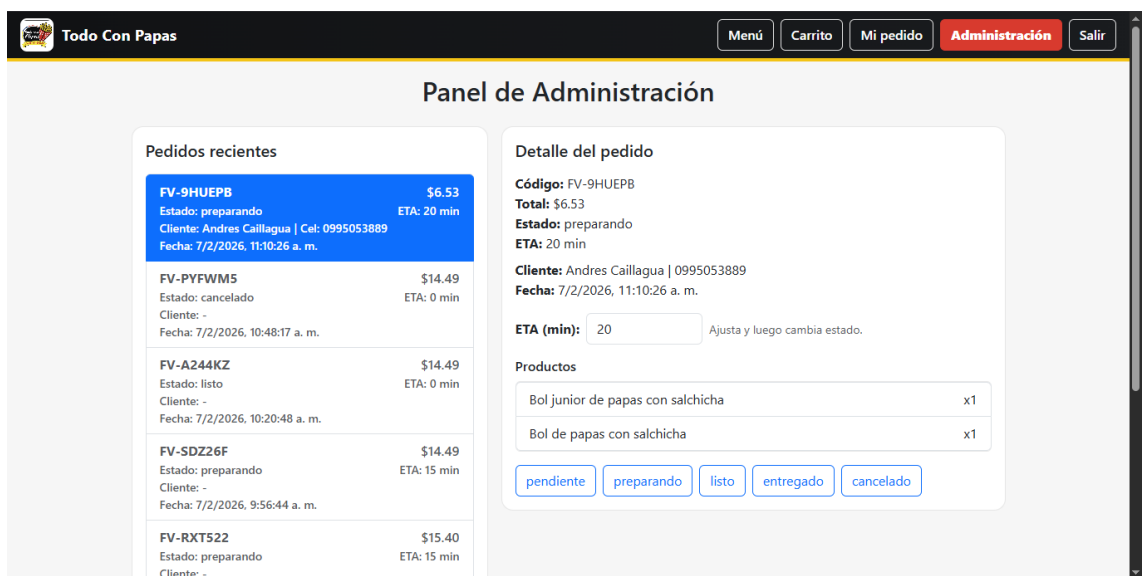


Figura 5: Panel de administración para gestión de pedidos

La figura muestra el panel de administración del sistema FilaVirtual, desde donde el personal puede gestionar pedidos, actualizar estados y controlar los tiempos estimados de atención.

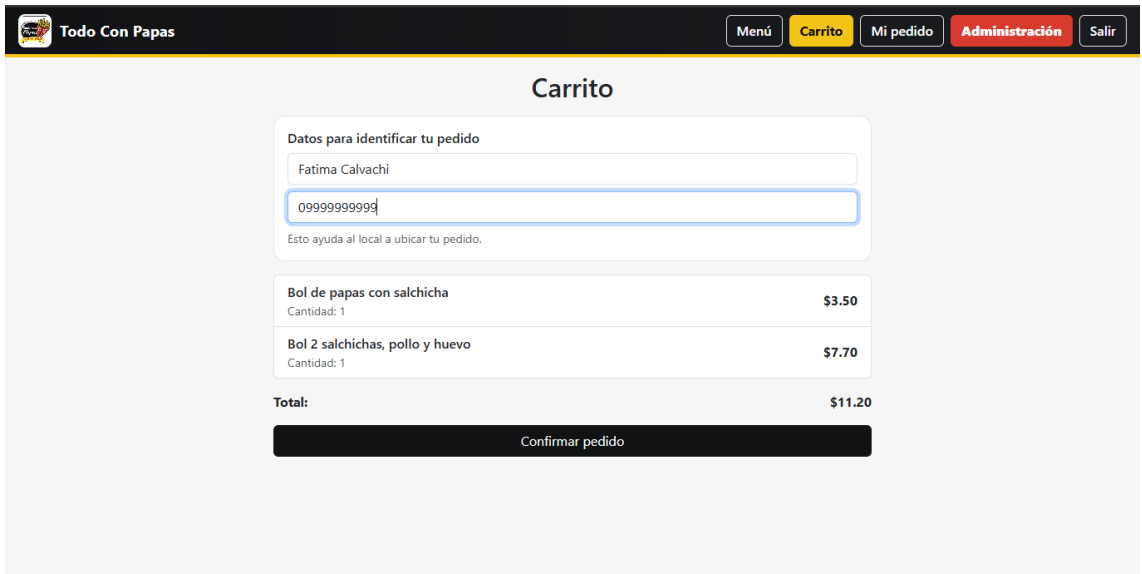
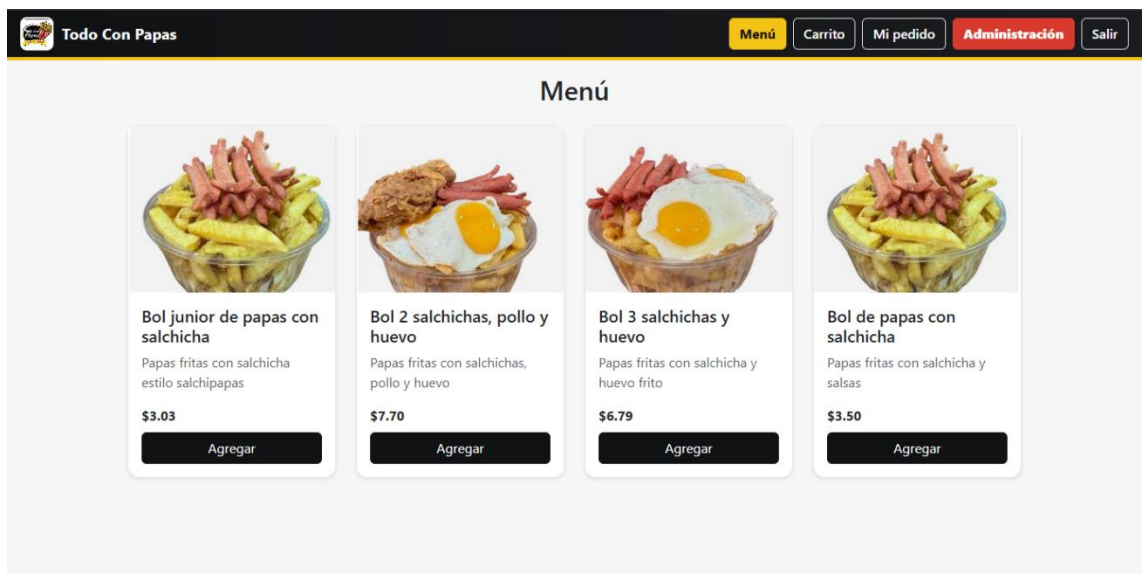



Figura 6: Interfaz del carrito de compras

Interfaz del carrito de compras previo a la confirmación del pedido.

Anexos

ANEXO A. Capturas del sistema FilaVirtual




Todo Con Papas

[Menú](#)
[Carrito](#)
[Mi pedido](#)
[Administración](#)
[Salir](#)

Carrito

Datos para identificar tu pedido


Fatima Calvachi

0995053389

Esto ayuda al local a ubicar tu pedido.

Bol de papas con salchicha	\$3.50
<small>Cantidad: 1</small>	
Bol 2 salchichas, pollo y huevo	\$7.70
<small>Cantidad: 1</small>	
Total:	\$11.20

Confirmar pedido


Todo Con Papas

[Menú](#)
[Carrito](#)
[Mi pedido](#)
[Administración](#)
[Salir](#)

Mi Pedido

Código del pedido


FV-HS2DVG Ver

Tip: el código se genera al confirmar el pedido en Carrito.

Código: FV-HS2DVG
Estado: pendiente
Tiempo estimado: 15 min
Total: \$11.20

Productos

Bol de papas con salchicha	x1
Bol 2 salchichas, pollo y huevo	x1


Todo Con Papas

[Menú](#)
[Carrito](#)
[Mi pedido](#)
[Administración](#)

Acceso Trabajador

Usuario

Contraseña

Ingresar

Demo: usuario **admin** / contraseña **1234** (o lo que tengas en .env)

Todo Con Papas

Menú
Carrito
Mi pedido
Administración
Salir

Panel de Administración

Pedidos recientes

FV-HS2DVG	\$11.20
Estado: pendiente	ETA: 15 min
Cliente: Fatima Calvachi Cel: 0995053389	
Fecha: 7/2/2026, 12:00:28 p. m.	
FV-9HUEPB	\$6.53
Estado: preparando	ETA: 20 min
Cliente: Andres Callagua Cel: 0995053889	
Fecha: 7/2/2026, 11:10:26 a. m.	
FV-PYFWM5	\$14.49
Estado: cancelado	ETA: 0 min
Cliente: -	
Fecha: 7/2/2026, 10:48:17 a. m.	
FV-A244KZ	\$14.49
Estado: listo	ETA: 0 min
Cliente: -	
Fecha: 7/2/2026, 10:20:48 a. m.	
FV-SDZ26F	\$14.49
Estado: preparando	ETA: 15 min
Cliente: -	

Detalle del pedido

Código: FV-HS2DVG
Total: \$11.20
Estado: pendiente
ETA: 15 min
Cliente: Fatima Calvachi | 0995053389
Fecha: 7/2/2026, 12:00:28 p. m.

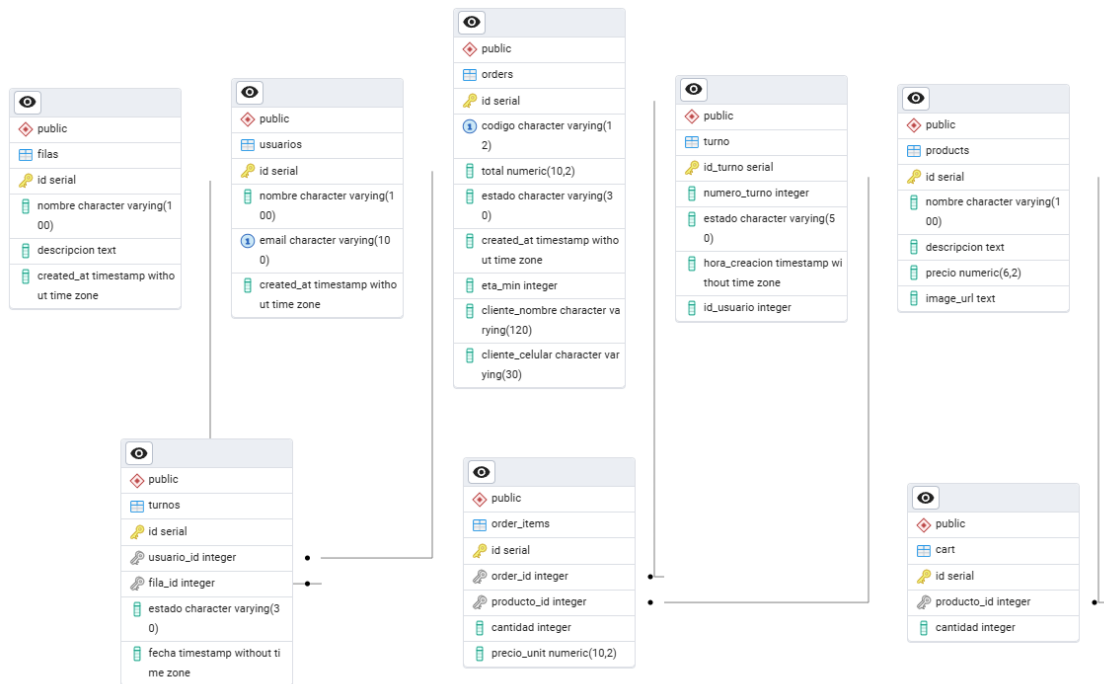
ETA (min): Ajusta y luego cambia estado.

Productos

Bol de papas con salchicha	x1
Bol 2 salchichas, pollo y huevo	x1

pendiente
preparando
listo
entregado
cancelado

ANEXO B. Modelo de Base de Datos



Nota: El Anexo presenta el modelo de datos general propuesto para el sistema FilaVirtual. Para la implementación funcional del sistema se prioriza el módulo de gestión de pedidos, representado principalmente por las tablas orders, order_items, products y cart. Las demás tablas corresponden a módulos proyectados o extensiones futuras del sistema.

ANEXO C – Configuración del Backend

```
backend > src > JS appjs > ...
17  const app = express();
18  |
19  app.use(
20  |   cors({
21  |     origin: "http://localhost:5173",
22  |     credentials: true,
23  |   })
24  | );
25  |
26  app.use(express.json());
27  |
28  //  rutas
29  app.use("/api/usuarios", usuarioRoutes);
30  app.use("/api/reservas", reservaRoutes);
31  app.use("/api", productsRoutes);
32  app.use("/api/carrito", cartRoutes);
33  app.use("/api/auth", authRoutes);
34  app.use("/api/turnos", turnosRoutes);
35  app.use("/api/orders", orderRoutes);
36  |
37  //  endpoints base
38  app.get("/", (req, res) => res.json({ message: "FilaVirtual Backend OK" }));
39  app.get("/status", (req, res) => res.json({ status: "ok", time: new Date() }));
40  |
41  //  Socket.io con HTTP server
42  const server = http.createServer(app);
43  |
44  const io = new Server(server, {
45  |   cors: {
46  |     origin: "http://localhost:5173",
47  |     credentials: true,
48  |   },
49  | });
50  |
```

Este anexo muestra la configuración principal del servidor backend del sistema FilaVirtual, incluyendo la inicialización de Express y la integración de Socket.io para comunicación en tiempo real.

ANEXO D – Lógica de Gestión de Pedidos (Backend)

```
export async function checkout(req, res) {
  try {
    const { cliente_nombre, cliente_celular } = req.body || {};

    if (!cliente_nombre?.trim() || !cliente_celular?.trim()) {
      return res.status(400).json({ message: "Falta nombre/apellido o celular" });
    }

    const result = await createOrderFromCart({
      cliente_nombre: cliente_nombre.trim(),
      cliente_celular: cliente_celular.trim(),
    });

    if (result.error) return res.status(400).json({ message: result.error });
  }
}
```

Este anexo presenta la lógica principal para la creación y actualización de pedidos, donde se asigna un código único al pedido y se notifican los cambios en tiempo real mediante Socket.io.

ANEXO E – Interfaz del Usuario (Frontend)

```
return (
  <div className="mx-auto" style={{ maxWidth: 720 }}>
    <h2 className="text-center mb-3">Carrito</h2>

    {error && <div className="alert alert-danger">{error}</div>}

    {codigo && (
      <div className="alert alert-success">
        <div><b>✔ Pedido creado</b></div>
        <div>Tu código para pagar en caja es:</div>
        <h4 className="mt-2 mb-0">{codigo}</h4>
        <div className="small text-muted mt-2">
          En "Mi pedido" verás el estado en tiempo real.
        </div>
      </div>
    )}

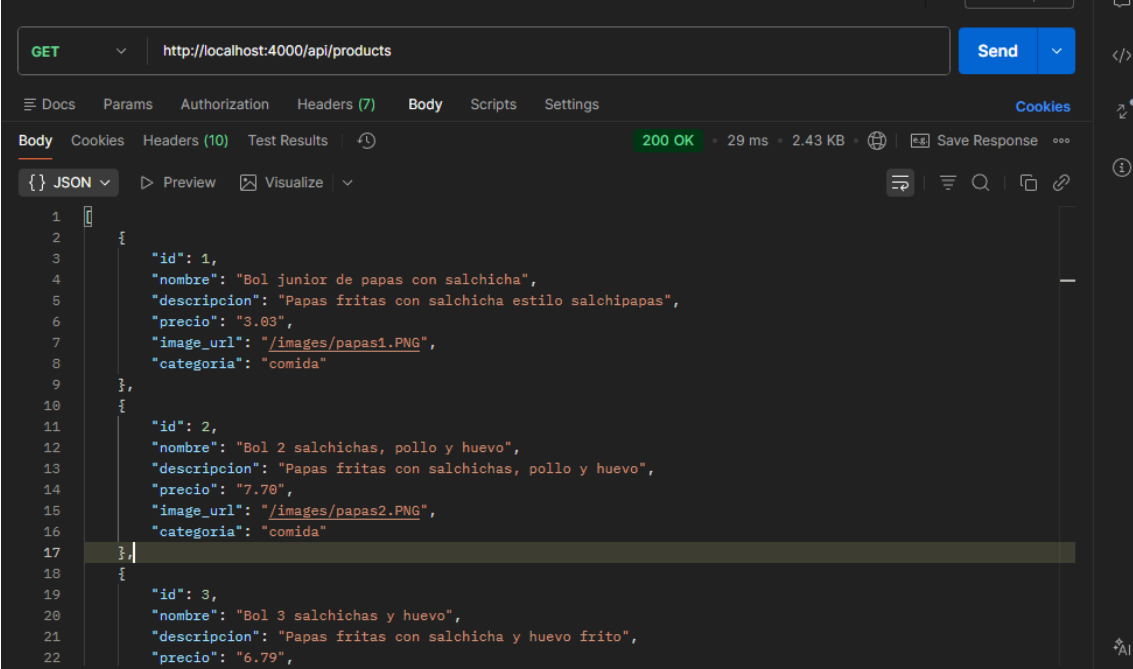
    {cargando ? (
      <p className="text-center">Cargando carrito...</p>
    ) : items.length === 0 ? (
      <p className="text-center text-muted">Tu carrito está vacío.</p>
    ) : (
      <>
        <div className="card p-3 mb-3">
          <h6 className="mb-2">Datos para identificar tu pedido</h6>

          <input
            className="form-control mb-2"
            placeholder="Nombre y apellido"
            value={clienteNombre}
            onChange={(e) => setClienteNombre(e.target.value)}
          />
        </div>
      </>
    )}
  </div>
)
```

Dentro de este anexo se incluyen varias capturas y bloques de código del frontend desarrollado con React. El objetivo es mostrar de forma técnica cómo estructuré el proceso de creación de pedidos y la manera en que el sistema gestiona la entrada de datos del usuario para finalizar la compra correctamente.

ANEXO F – Pruebas funcionales del sistema

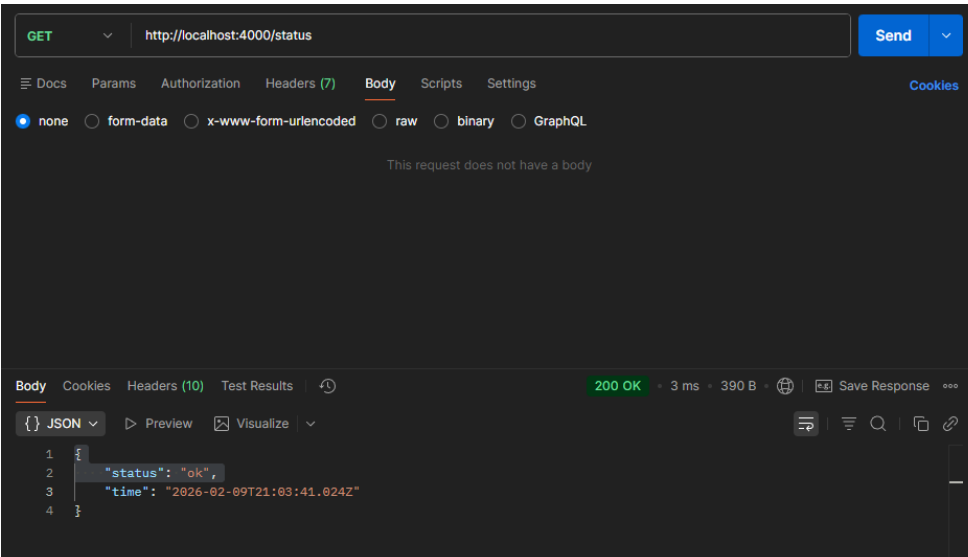
Anexo F.1 – Consulta de productos



```
GET http://localhost:4000/api/products 200 OK • 29 ms • 2.43 KB
Body
JSON
1 {
2   "id": 1,
3   "nombre": "Bol junior de papas con salchicha",
4   "descripcion": "Papas fritas con salchicha estilo salchipapas",
5   "precio": "3.03",
6   "image_url": "/images/papas1.PNG",
7   "categoria": "comida"
8 },
9
10 {
11   "id": 2,
12   "nombre": "Bol 2 salchichas, pollo y huevo",
13   "descripcion": "Papas fritas con salchichas, pollo y huevo",
14   "precio": "7.70",
15   "image_url": "/images/papas2.PNG",
16   "categoria": "comida"
17 },
18
19 {
20   "id": 3,
21   "nombre": "Bol 3 salchichas y huevo",
22   "descripcion": "Papas fritas con salchicha y huevo frito",
23   "precio": "6.79",
```

Nota: Se muestra la consulta del menú de productos desde el backend, donde se obtiene la lista registrada en formato JSON.

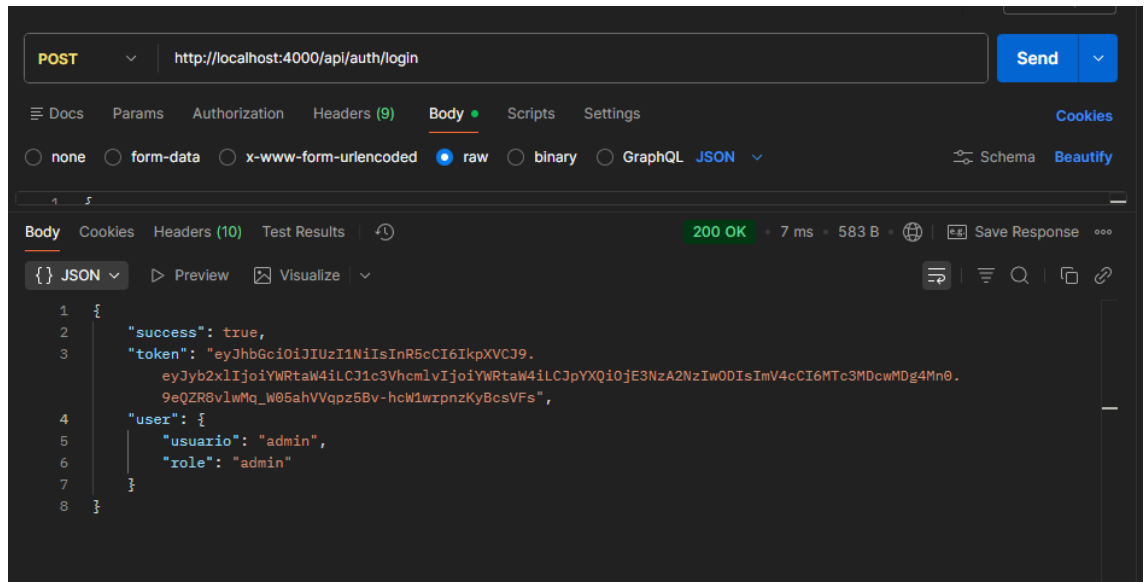
Anexo F.2 – Verificación del estado del backend



```
GET http://localhost:4000/status 200 OK • 3 ms • 390 B
Body
JSON
1 {
2   "status": "ok",
3   "time": "2026-02-09T21:03:41.024Z"
4 }
```

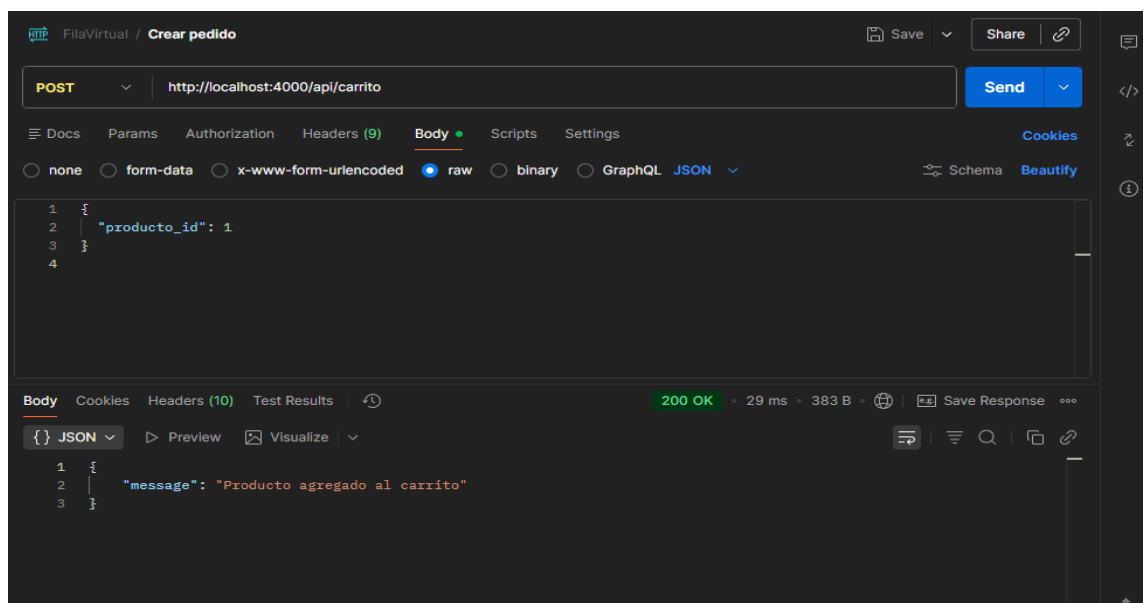
Nota: Se presenta la verificación del estado del backend, confirmando que el sistema se encuentra activo y disponible para recibir solicitudes.

Anexo F.3 – Autenticación de administrador



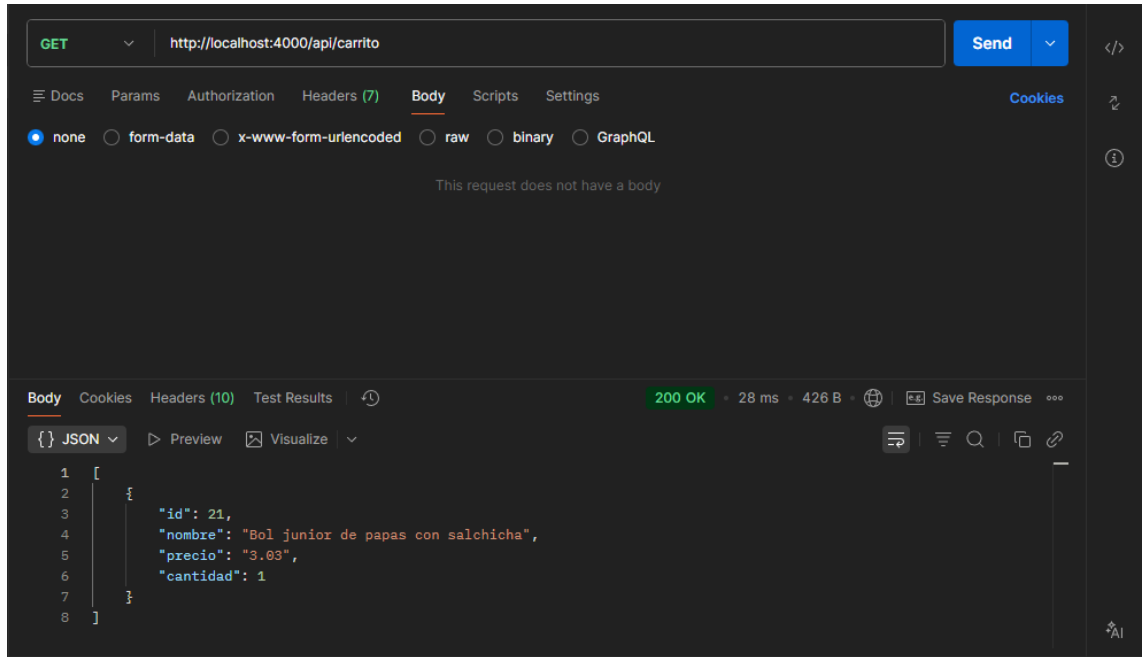
Nota: Se valida el proceso de inicio de sesión, obteniendo un token JWT utilizado para acceder a endpoints protegidos del sistema.

Anexo F.4 – Agregar producto al carrito



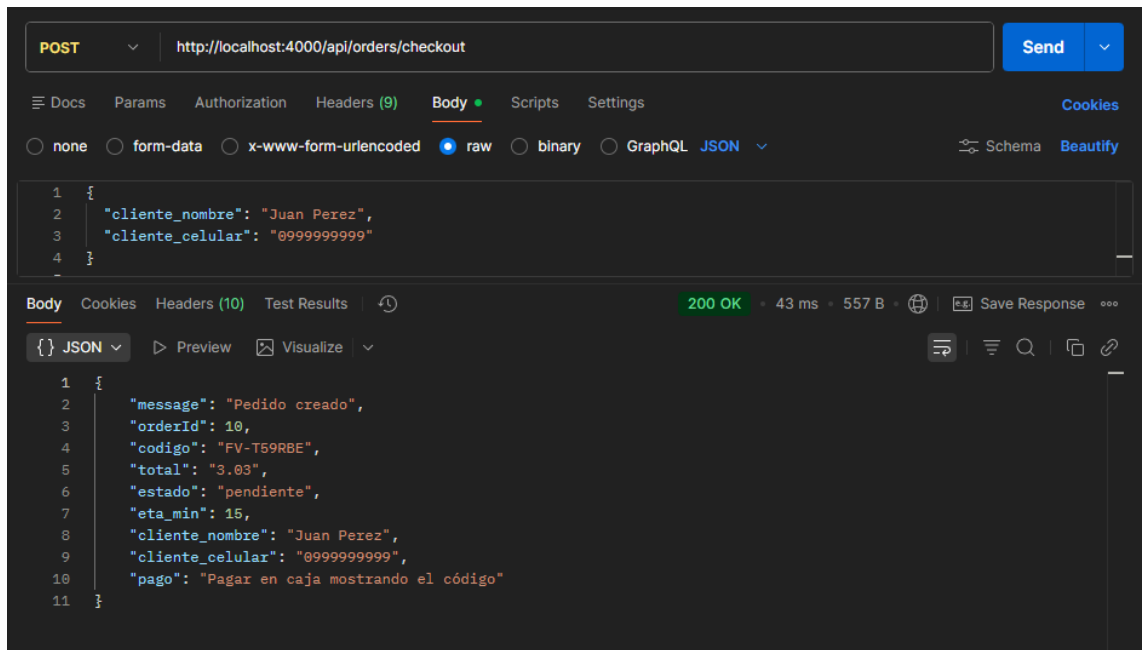
Nota: La prueba confirma la correcta inserción de un producto en el carrito de compras, almacenando el identificador del producto y su cantidad.

Anexo F.5 – Consulta del carrito



Nota: En esta captura se comprueba que el carrito procesa los artículos sin errores, listando cada producto con su nombre, el precio actualizado y la cantidad exacta que el usuario seleccionó.

Anexo F.6 – Creación de pedido



The screenshot shows a REST client interface with the following details:

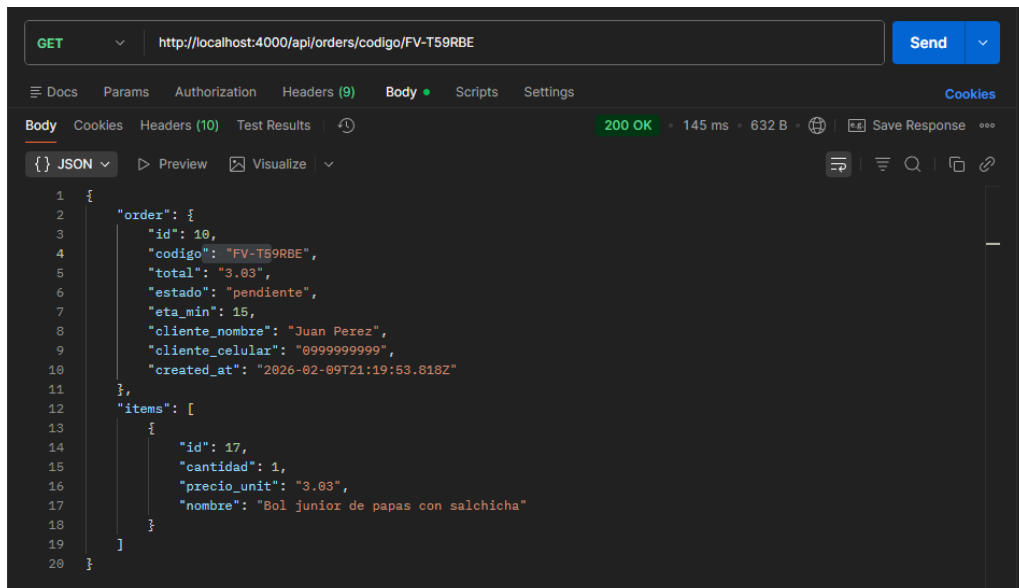
- Method:** POST
- URL:** http://localhost:4000/api/orders/checkout
- Body (raw):**

```
1 {
2   "cliente_nombre": "Juan Perez",
3   "cliente_celular": "0999999999"
4 }
```
- Status:** 200 OK
- Response (JSON):**

```
1 {
2   "message": "Pedido creado",
3   "orderId": 10,
4   "codigo": "FV-T59RBE",
5   "total": "3.03",
6   "estado": "pendiente",
7   "eta_min": 15,
8   "cliente_nombre": "Juan Perez",
9   "cliente_celular": "0999999999",
10  "pago": "Pagar en caja mostrando el código"
11 }
```

Nota: Al confirmar la orden, se crea un código de pedido único que vincula los datos del cliente y genera el ticket necesario para gestionar el pago en la caja del local.

Anexo F.7 – Consulta de pedido por código



The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:4000/api/orders/codigo/FV-T59RBE
- Status:** 200 OK
- Response (JSON):**

```
1 {
2   "order": {
3     "id": 10,
4     "codigo": "FV-T59RBE",
5     "total": "3.03",
6     "estado": "pendiente",
7     "eta_min": 15,
8     "cliente_nombre": "Juan Perez",
9     "cliente_celular": "0999999999",
10    "created_at": "2026-02-09T21:19:53.818Z"
11  },
12  "items": [
13    {
14      "id": 17,
15      "cantidad": 1,
16      "precio_unit": "3.03",
17      "nombre": "Bol junior de papas con salchicha"
18    }
19  ]
20 }
```

Nota: Con esta prueba verifiqué que el sistema recupera correctamente el pedido mediante su código, mostrando al instante el estado, el tiempo de espera y el detalle de los productos.