



UNIVERSIDAD VERACRUZANA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

PROGRAMA EDUCATIVO:

Ingeniería Informática

EXPERIENCIA EDUCATIVA:

Ingeniería de Software

DOCENTE:

Yadira Fleitas Toranzo

ESTUDIANTE(S):

Eduardo Arévalo Sánchez

Luis Emilio Baldwin Hipólito

ACTIVIDAD:

Sprint 3

LOCALIDAD:

Veracruz, Ver

FECHA:

13/12/2025



Sprint Planning: Sprint 3. Lógica de Negocio y CRUD

Objetivo del Sprint (Sprint goal): Implementar módulos de generación de reportes, herramientas de búsqueda y lógica avanzada para complementar la funcionalidad del sistema.

Trello Design Sprint Planning. Sprint 3

LinkTrello: <https://trello.com/invite/b/6937229ff57edf179dd40824/ATTI1bf19361d59a6d4886aab26ea52ec84c18E0DFF1/goorderapp>

ID	Historia de Usuario	Estimación	Prioridad
HU-07	Como cliente, quiero agregar, modificar o eliminar productos de mi carrito de compras para preparar mi pedido antes de confirmarlo.	6	Alta
HU-08	Como cliente, quiero guardar mi dirección de entrega para no tener que escribirla cada vez que hago un pedido.	5	Media
HU-09	Como personal del restaurante, quiero que el sistema local detecte automáticamente los nuevos pedidos web para procesarlos de inmediato.	13	Alta

Estimaciones Story Points (Planning Poker):

- HU-07. 6 puntos** Implica gestión de estado en el Frontend (State Management) para manejar cantidades y actualizaciones dinámicas del DOM sin recargar la página.
- HU-08. 5 puntos** Requiere crear una tabla tb_direcciones_usuarios relacionada por Foreign Key con el usuario.
- HU-09. 13 puntos** Esta funcionalidad requiere comunicación asíncrona avanzada. No es una simple consulta.



Sprint Backlog

HU-07: Carrito de Compras. Responsabilidad: Eduardo

Sprint Backlog ▾

✓ HU-07: Carrito de Compras

+ Añadir Checklist Adjunto Ubicación

Miembros Etiquetas
EA + Prioridad Alta Story Points: 6 +

Fechas
8 dic - 10 dic, 23:59 Cumplida ▾

≡ Descripción Editar

Como cliente, quiero agregar, modificar o eliminar productos de mi carrito de compras para preparar mi pedido antes de confirmarlo.

Criterios de Aceptación:

- El ícono del carrito en la barra de navegación debe mostrar un contador con la cantidad total de artículos.
- Al dar clic en "Agregar" en un producto, debe aparecer un feedback visual (ej. "Agregado al carrito")
- En la vista del carrito, debo poder sumar (+) o restar (-) unidades de cada producto.
- El subtotal y total deben recalcularse en tiempo real al modificar cantidades.

Tareas Técnicas:

- Frontend: Implementación para manejar el estado global del carrito
- Frontend: Crear lógica para guardar el carrito en localStorage (para no perderlo si recarga la página).
- Frontend: Maquetar la pantalla de "Resumen de Carrito" (Lista de ítems, precios, total).



HU-08: Gestión de Direcciones. Responsabilidad: Eduardo

Sprint Backlog ▾

Speaker icon | Filter icon ...

✓ HU-08: Gestión de Direcciones

+ Añadir Checklist Adjunto Ubicación

Miembros Etiquetas

EA +

Prioridad Media

Story Points: 5

+

Fechas

9 dic - 11 dic, 23:00 Cumplida ▾

☰ Descripción

Editar

Como cliente, quiero guardar mi dirección de entrega para no tener que escribirla cada vez que hago un pedido.

Criterios de Aceptación:

- El formulario debe validar campos obligatorios: Calle, Número, Colonia y Código Postal
- Debe permitir guardar la dirección con un alias (ej. "Casa", "Oficina").
- Si el usuario ya tiene direcciones guardadas, debe mostrarlas en una lista para seleccionar una.
- Si el usuario ya tiene direcciones guardadas, debe mostrarlas en una lista para seleccionar una.
- Debe permitir editar o eliminar una dirección existente.

Tareas Técnicas:

- Base de Datos: Crear tabla tb_cliente_direccion vinculada al ID del usuario.
- Backend: Crear endpoint POST /api/addresses para guardar nueva dirección.
- Backend: Crear endpoint GET /api/addresses para listar las direcciones del usuario logueado.
- Frontend: Crear formulario de dirección con validaciones.
- Frontend: Integrar la selección de dirección en el flujo de compra.



HU-09: Confirmación de Pedido. Responsabilidad: Eduardo

Sprint Backlog ▾

+

✓ HU-09: Confirmación de Pedido

+ Añadir Checklist Adjunto Ubicación

Miembros Etiquetas

EA + Prioridad Alta Story Points: 8 +

Fechas

9 dic - 13 dic, 23:00 Cumplida ▾

≡ Descripción Editar

Como cliente, quiero confirmar mi pedido final para que sea enviado al restaurante y empiecen a prepararlo.

Criterios de Aceptación:

- Cuando se crea un pedido en la Web, debe aparecer en la pantalla de "Pedidos" del sistema de escritorio sin necesidad de reiniciar.
- La información (platillos y notas) debe ser idéntica en ambos sistemas.
- Al confirmar, se debe mostrar una pantalla de éxito con el Número de Pedido.
- El pedido al completarse debe aparecer en historial de pedidos

Tareas Técnicas:

- Base de Datos: Asegurar que ambos sistemas apunten a la misma IP/Instancia de MySQL
- Escritorio (Java/Netbeans): Implementar un Timer o Hilo que consulte la tabla tb_pedido cada 30 segundos (Polling) buscando estado "NUEVO".
- Pruebas: Realizar un pedido en Web y cronometrar cuánto tarda en aparecer en Escritorio.



Daily Scrum

DAILY SCRUM				
	EDUARDO		EMILIO	
LO QUE HICE AYER	Crear tabla tb_direccion en BD.	Endpoint POST para direcciones listo.	Configurar conexión JDBC en Escritorio	Diseñar el carrito
LO QUE HARÉ HOY	Programar Timer en Java	Endpoint GET pedidos nuevos	Probar sincronización local.	Formulario dirección Calcular el total
MIS CONTRATIEMPOS	Latencia en BD remota.	Ninguno	Fallo de conexión JDBC.	Error visual. Datos no guardaban.



Sprint Review: Sprint 3

Fecha fin del Sprint: 13/12/2025

Se ha cumplido el Sprint Goal. La gestión del flujo de compra es funcional, permitiendo a los usuarios administrar su carrito, guardar direcciones de entrega y confirmando que los pedidos se sincronizan exitosamente con el sistema local del restaurante

Historias Completadas vs Planeadas

ID	Historias de Usuario	Estado
HU-07	Carrito de Compras	(Completado)
HU-08	Gestión de Direcciones	(Completado)
HU-09	Confirmación de Pedido	(Completado)

Incremento Funcional:

En este Sprint se entrega la **versión funcional 3.0** de la aplicación web 'GoOrder', integrando la lógica avanzada de negocio y la comunicación entre sistemas.

Carrito de Compras Interactivo: Se implementó la gestión de estado global en el Frontend, permitiendo a los usuarios agregar productos, modificar cantidades y visualizar el recálculo automático de subtotales y totales en tiempo real sin recargar la página

Administración de Direcciones de Entrega: Se habilitó el módulo de perfil de usuario para registrar y gestionar direcciones. El sistema ahora valida campos obligatorios (Calle, CP, Colonia) y almacena la información en la base de datos vinculándola al cliente para agilizar futuros pedidos

Sincronización Web-Restaurante: Se logró la comunicación asíncrona entre el entorno Web y el sistema de Escritorio. Ahora, cuando un cliente confirma un pedido en la web, el sistema local lo detecta automáticamente (mediante *Polling*) y lo despliega en la pantalla de cocina para su preparación inmediata



Sprint Backlog

→ ← ⋮

HU-07: Carrito de Compras
⌚ 8 dic - 10 dic ⚒ 7/7 EA

HU-08: Gestión de Direcciones
⌚ 9 dic - 11 dic ⚒ 9/9 EA

HU-09: Confirmación de Pedido
⌚ 9 dic - 13 dic ⚒ 7/7 EA

+ Añade una tarjeta



Evidencias

1. Captura de las compras

Menú GoOrder

Hola, ¿qué se te antoja hoy?

The screenshot shows the main menu interface. At the top, there are category filters: 'Todos' (highlighted in orange), 'Entradas', 'Platos Fuertes', 'Bebidas', and 'Postres'. Below the filters are three cards representing menu items:

- Alitas de pollo a la barbacoa \$180**
Platillo tradicional bañado en salsa BBQ. Opción ideal para compartir.
- Corte New York \$350**
Corte fino asado al carbón con guarnición de vegetales.
- Hamburguesa Clásica \$120**
Carne de res 100%, lechuga, tomate y queso cheddar.

Menú GoOrder

Hola, Eduardo Arévalo Sánchez

The screenshot shows the menu page with a shopping cart overlay. The cart contains the following items:

Plato	Precio
Corte New York	\$350
Hamburguesa Clásica	\$120
Alitas de pollo a la barbacoa	\$180
Mixología de Autor	\$95

Total: \$745.00

Comprar Ahora



2. Captura del registro de la dirección

Menú GoOrder

Hola, Eduardo Arévalo Sánchez

Seguir comprando

Finalizar Compra

Resumen del Pedido

Corte New York	\$350
Hamburguesa Clásica	\$120
Ailitas de pollo a la barbacoa	\$180
Mixología de Autor	\$95

Total a pagar: **\$745.00**

Datos de Envío y Pago

Dirección de entrega
Mercaderías #481, Col. Dos caminos

Número de Tarjeta
0000 0000 0000 0000

Expira (MM/AA) CVV
12/25 123

Pagar Pedido

3. Captura de la confirmación del pedido.

Menú GoOrder

Hola, Eduardo Arévalo Sánchez

Seguir comprando

Finalizar Compra

Resumen del Pedido

Corte New York	\$350
Hamburguesa Clásica	\$120
Ailitas de pollo a la barbacoa	\$180
Mixología de Autor	\$95

Total a pagar: **\$745.00**

Datos de Envío y Pago

Dirección de entrega
Mercaderías #481, Col. Dos caminos

Número de Tarjeta
1231412414124141

Expira (MM/AA) CVV
12/28 ...

Pagar Pedido

¡Pedido realizado con éxito! ID: 3

Aceptar



Feedback del Product Owner

- Es funcional ver cómo un pedido realizado en la web aparece automáticamente en el sistema de escritorio sin intervención manual
- La gestión del carrito es intuitiva; los cálculos de totales son inmediatos y facilitan la decisión de compra del usuario.
- El formulario de direcciones valida bien los datos y agiliza el proceso de checkout al permitir seleccionar ubicaciones guardadas

SprintRetrospective: Sprint 3

¿Qué funcionó bien?

- La implementación de CartContext en el Frontend permitió manejar el estado del carrito de compras de manera global y limpia sin "prop drilling" excesivo.
- La estrategia de Polling mediante hilos en Java resultó efectiva para detectar los pedidos nuevos sin complejizar la arquitectura con WebSockets
- El diseño de la base de datos (tb_cliente_direccion) permitió escalar la relación de usuarios a múltiples direcciones sin redundancia

¿Qué no funcionó?

- Tuvimos dificultades con la persistencia de datos en **localStorage**; al principio, el carrito se vaciaba si el usuario refrescaba la página inesperadamente
- Hubo problemas de latencia y bloqueos momentáneos en la interfaz de escritorio (Java) cuando el hilo de conexión a la BD tardaba en responder
- La coordinación de los IDs de estado del pedido ("Nuevo", "En Proceso") generó confusión entre el equipo al inicio del sprint.

Acciones de Mejora

1. **Acción:** Implementar un "Loader" o indicador visual en el escritorio para que el personal sepa cuando el sistema está buscando pedidos nuevos, evitando que crean que se trabó. Responsable: Eduardo (Backend/Escritorio)
2. **Acción:** Refactorizar el manejo de localStorage creando un Custom Hook reutilizable para asegurar que los datos del usuario (carrito y sesión) estén siempre sincronizados. Responsable: Luis Emilio (Frontend)



Retroespective

→ ← ⋮

Lo que Funcionó: Sprint 3

☰

Lo que no Funcionó: Sprint 3

☰

Acciones de Mejora: Sprint 3

☰

Lo que Funcionó: Sprint 2

☰

Lo que no Funcionó: Sprint 2

☰

+ Añade una tarjeta