



UNIVERSIDAD VERACRUZANA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

PROGRAMA EDUCATIVO:

Ingeniería Informática

EXPERENCIA EDUCATIVA:

Ingeniería de Software

DOCENTE:

Yadira Fleitas Toranzo

ESTUDIANTE(S):

Eduardo Arévalo Sánchez

Luis Emilio Baldwin Hipólito

ACTIVIDAD:

Sprint 4

LOCALIDAD:

Veracruz, Ver

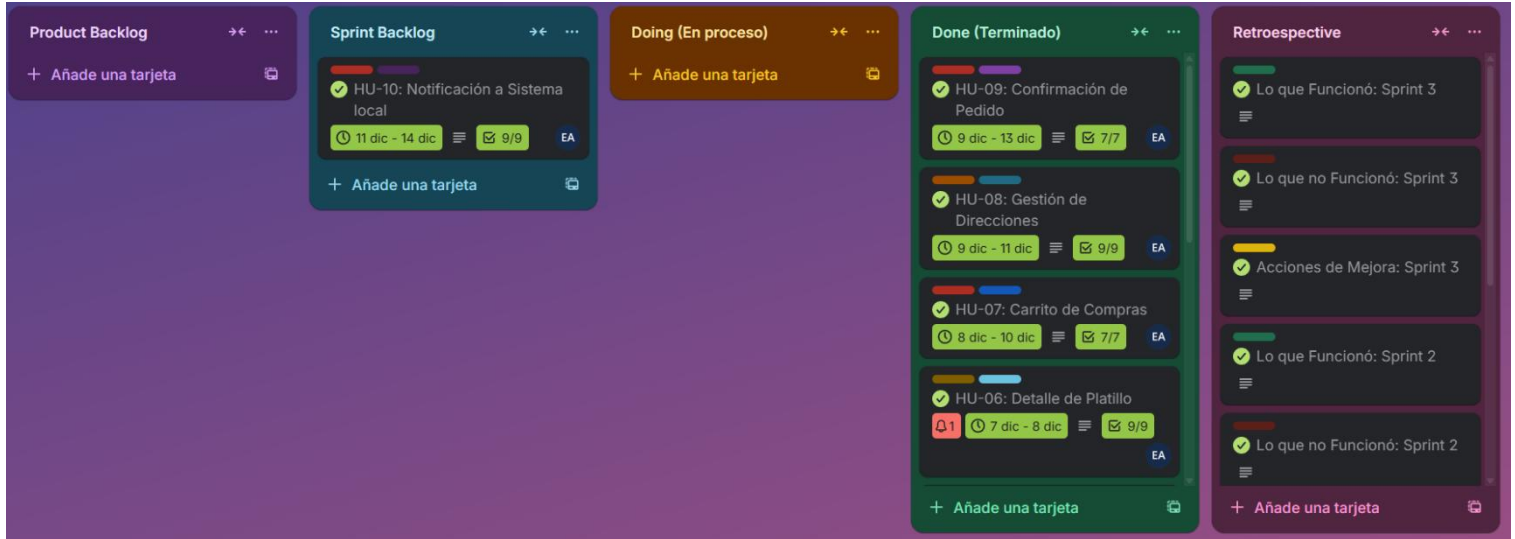
FECHA:

08/12/2025



Sprint Planning: Sprint 4. Lógica de Negocio y CRUD

Objetivo del Sprint (Sprint goal): Finalizar el sistema implementando las notificaciones en tiempo real, realizar pruebas integrales (Depuración) y preparar la entrega final del producto.



Trello Design Sprint Planning. Sprint 4

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

ID	Historia de Usuario	Estimación	Prioridad
HU-10	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-10. 13 puntos** debido a la complejidad técnica de implementar hilos de ejecución en segundo plano en Java para el monitoreo constante de la base de datos y la sincronización en tiempo real sin congelar la interfaz.



Sprint Backlog

HU-10: Notificación a Sistema local. Responsabilidad: Eduardo

Sprint Backlog ▾

🔊 | 🖼️ | ⋮ | ✕

✅ **HU-10: Notificación a Sistema local**

+ Añadir | 📋 Checklist | 📎 Adjunto | 📍 Ubicación

Miembros Etiquetas

EA + Prioridad Alta Story Points: 13 +

Fechas

11 dic - 14 dic, 23:00 Cumplida ▾

☰ Descripción Editar

Como personal del restaurante, quiero que el sistema local detecte automáticamente los nuevos pedidos web para procesarlos de inmediato.

Criterios de Aceptación:







- La pantalla de confirmación debe mostrar el resumen final: Productos, Dirección de envío y Total a Pagar.
- El botón "Confirmar Pedido" debe estar bloqueado si no hay dirección seleccionada.
- Al confirmar, se debe mostrar una pantalla de éxito con el Número de Pedido.
- El carrito debe vaciarse automáticamente después de una compra exitosa.

Tareas Técnicas:

- Base de Datos: Verificar que existan las tablas de pedidos (tb_pedido y tb_detalle_pedido) en el sistema actual.
- Backend: Crear endpoint POST /api/orders que reciba el carrito y la dirección.
- Backend: Implementar transacción SQL (Insertar en cabecera tb_pedido y luego sus renglones en tb_detalle_pedido).
- Frontend: Conectar botón de pago con el endpoint.
- Frontend: Crear página de "Gracias por tu compra" (Success Page).



Daily Scrum

DAILY SCRUM					
	EDUARDO			EMILIO	
LO QUE HICE AYER	Programar Timer Polling.	Consulta SQL pedidos.	Detectar estado "Nuevo".	Fix bug carrito.	Ajuste CSS móvil. 
LO QUE HARÉ HOY	Integrar alerta	Aviso visual (Pop-up).	Probar tiempos respuesta. 	Prueba estrés web.	Redactar Manuales.
MIS CONTRATIEMPOS	Ninguno 	Latencia en red.	Ninguno	Conflicto Merge Git.	Caché no limpiaba.
 ¡Sí!  					



Sprint Review: Sprint 4

Fecha fin del Sprint: 14/12/2025

Se ha cumplido el Sprint Goal. El sistema ha alcanzado su versión final estable. Se completó la integración entre la plataforma web y el sistema local mediante notificaciones automáticas, y se realizaron las pruebas exhaustivas de depuración para asegurar un flujo de venta sin errores.

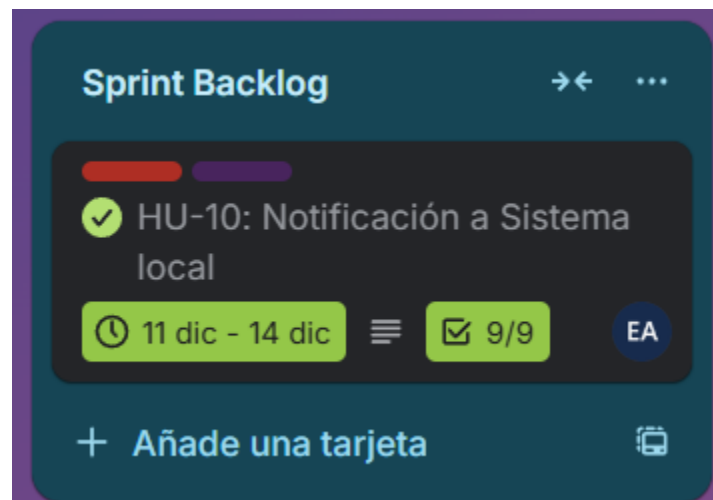
Historias Completadas vs Planeadas

ID	Historias de Usuario	Estado
HU-10	Notificación a Sistema local	(Completado)

Incremento Funcional:

En este Sprint se entrega la versión final 4.0 de la aplicación 'GoOrder', completando el ciclo de vida del desarrollo. El incremento se centra en la conectividad total y la estabilidad del sistema.

Sistema de Notificaciones en Tiempo Real: Se implementó la lógica de escucha activa en la aplicación de escritorio. Ahora, el personal del restaurante recibe alertas visuales automáticas en el momento exacto en que un cliente confirma un pedido en la web, eliminando la necesidad de recargar la pantalla manualmente.





GoOrder-APP

Ingeniería de Software



Evidencias

1. Captura del Centro de Notificaciones

Interfaz

Usuario Menú Producto Cliente Categoría Reportes Cerrar Sesión

Centro de Notificaciones - Pedidos Web

Pedidos Pendientes:

ID Pedido	Cliente	Fecha	Dirección	Total	Estado
2	Eduardo Arévalo Sánchez	2025-12-13 17:51:24.0	Mercadelas #481, Col. Dos c.	\$420.0	PENDIENTE
1	Eduardo Arévalo Sánchez	2025-12-13 00:44:43.0	Mercadelas #481, Col. Dos c.	\$350.0	PENDIENTE

Message

¡ Pedido actualizado correctamente.

OK

Detalle del Pedido Seleccionado:

Producto	Cantidad	Precio Unit.	Subtotal
Hamburguesa Clásica	1	\$120.0	\$120.0
Hamburguesa Clásica	1	\$120.0	\$120.0
Alitas de pollo a la barbacoa	1	\$180.0	\$180.0

Actualizar Lista Despachar Pedido

Centro de Notificaciones - Pedidos Web

Pedidos Pendientes:

ID Pedido	Cliente	Fecha	Dirección	Total	Estado
3	Eduardo Arévalo Sánchez	2025-12-14 00:23:28.0	Mercadelas #481, Col. Dos c.	\$745.0	PENDIENTE
2	Eduardo Arévalo Sánchez	2025-12-13 17:51:24.0	Mercadelas #481, Col. Dos c.	\$420.0	PENDIENTE
1	Eduardo Arévalo Sánchez	2025-12-13 00:44:43.0	Mercadelas #481, Col. Dos c.	\$350.0	PENDIENTE

Detalle del Pedido Seleccionado:

Producto	Cantidad	Precio Unit.	Subtotal
Corte New York	1	\$350.0	\$350.0
Hamburguesa Clásica	1	\$120.0	\$120.0
Alitas de pollo a la barbacoa	1	\$180.0	\$180.0
Mixología de Autor	1	\$95.0	\$95.0

Actualizar Lista Despachar Pedido



Feedback del Product Owner

- La notificación visual y sonora cumple con el objetivo; el personal se entera de inmediato sin estar pegado a la pantalla.
- Tras las pruebas de estrés, la aplicación no se congela ni pierde la conexión con la base de datos.
- Se confirma que el ciclo completo (Cliente pide en Web \rightarrow Restaurante recibe en Escritorio \rightarrow Cocina prepara) funciona sin interrupciones.

SprintRetrospective: Sprint 4

¿Qué funcionó bien?

- La implementación del TimerTask en Java fue una solución eficiente para el polling de pedidos sin afectar el rendimiento del PC.
- La fase dedicada exclusivamente a Depuración nos permitió limpiar errores "invisibles" que arrastrábamos de sprints anteriores.
- La comunicación entre el equipo para realizar el Merge final a la rama main evitó que sobrescribiéramos código funcional.

¿Qué no funcionó?

- Tuvimos problemas de compatibilidad con la librería de audio (javax.sound) en algunas computadoras, lo que retrasó la alerta sonora.
- Subestimamos el tiempo necesario para redactar los Manuales de Usuario finales; se hizo con mucha prisa al final
- Surgieron conflictos de código inesperados al unir la rama del *Sprint 3* con la del Sprint 4 el último día.

Acciones de Mejora

1. Acción: Integrar y probar el código en la rama principal (main) al final de cada semana, no esperar hasta el final del sprint.
Responsable: Scrum Master / Tech Lead
2. Acción: Documentar las funcionalidades en el manual técnico *a medida* que se desarrollan, en lugar de dejarlo todo para la entrega final.
Responsable: Todos los miembros del equipo



GoOrder-APP

Ingeniería de Software



Retroespective

✓ Lo que Funcionó: Sprint 4

✓ Lo que no Funcionó: Sprint 4

✓ Acciones de Mejora: Sprint 4

✓ Lo que Funcionó: Sprint 3

✓ Lo que no Funcionó: Sprint 3

+ Añade una tarjeta