

}{[\*"Y&%GtNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Ingeniería de Software II

Semestre II – 2025



## Sprint 8

Integrantes:

Ciprian Jimenez, Mishell Rosa Elvira - 231169

Mejicanos Hernandez, Abner Gabriel - 231134

Nájera Marakovits, Ingrid Nina Alessandra - 231088

Ramírez Velásquez, Diego Alejandro - 23601

Rivera Rodriguez, Alejandro - 23674

Yee Vidal, María José - 231193

## Producto Backlog

- **Pila del producto**

Funcionalidad	Sprint en que se completó
Registro y gestión de clientes con historial de pedidos	Sprint 1
Control de stock de inventario	Sprint 1
Calculadora de materiales para pedidos	Sprint 1
Reportes de ventas con gráficas y filtros	Sprint 1
Implementación de login y registro	Sprint 1
Navegación entre vistas funcional	Sprint 2
Rediseño visual del sistema	Sprint 2
Vista de clientes y funcionalidad CRUD	Sprint 2
Tablas en backend para clientes, login y reportes	Sprint 2
Funcionalidades completas de proveedores	Sprint 3
Gestión de stock del inventario	Sprint 3
Conexión backend-frontend del inventario	Sprint 3
Compra y configuración del servidor	Sprint 3
Unificación del proyecto y despliegue en servidor	Sprint 3
Rediseño completo y responsivo	Sprint 4
APIs funcionales para contabilidad y proveedores	Sprint 4
Navegación completa mediante barra y botones de retorno	Sprint 4
Formularios y búsqueda funcional en inventario y proveedores	Sprint 4
Arreglo de errores en clientes y proveedores	Sprint 5
Agregar página de inicio para todo público	Sprint 5
Funcionalidad de la página de detalle de ventas	Sprint 5
Gráficas de resumen por fecha en detalle de ventas	Sprint 5
Gráficas del home funcionando con datos reales	Sprint 5
Arreglo para hacer peticiones tanto localmente como desde el servidor.	Sprint 5
Funcionalidad CRUD completa para ventas	Sprint 6
Vista de historial de ventas y formulario para nueva venta	Sprint 6
Métodos para modificar y eliminar ventas en backend	Sprint 6
Arreglo y mejora visual de la vista de clientes y proveedores	Sprint 6
Estilización uniforme del sistema (styles.css)	Sprint 6
Reconfiguración y relanzamiento del servidor	Sprint 6
Quitar herramientas de desarrollo (devtools) para producción	Sprint 6
Arreglo de errores con inventario	Sprint 6
Finalización y conexión de gráficas del home con datos reales	Sprint 6
Implementación de seguridad: restricción de acceso a páginas solo para usuarios loggeados	Sprint 7

Generación automática de facturas por cada transacción	Sprint 7
Descarga de facturas en formato PDF	Sprint 7
Mejoras de diseño en vistas (clientes, proveedores, inventario)	Sprint 7
Corrección de bugs en navegación y formularios	Sprint 7
Ajustes en el reporte de ventas (gráficas, filtros y tablas)	Sprint 7
Optimización de conexión backend–frontend (inventario y proveedores)	Sprint 7
Ajustes en configuración del servidor y despliegue	Sprint 7
Revisar qué problemas hay en el sistema y qué se debe corregir	Sprint 8
Clientes, clientes registro, proveedores, reporte de ventas	Sprint 8
Landing page, login, register, forgot password, contabilidad	Sprint 8
Funcionalidad al botón de enviar en landing page	Sprint 8
Implementar pinta en todas las vistas	Sprint 8
Eliminar tablas innecesarias	Sprint 8
Arreglar el serializer de compra	Sprint 8
Inventario, home, pedidos, billpage (estilos)	Sprint 8
Arreglar problemas con el historial de pedidos	Sprint 8
Pruebas de seguridad	Sprint 8
Pruebas de volumen	Sprint 8
Pruebas de carga	Sprint 8
Unificar los serializers de producto	Sprint 8
Arreglar la duplicación del dominio de órdenes	Sprint 8
Arreglar el modelo de la tabla de ventas	Sprint 8
Unificar las clases de ventas	Sprint 8
Cambiar diseño de login	Sprint 8
Unificar las clases duplicadas de telas	Sprint 8
Limpieza de imports (quitar duplicados)	Sprint 8
Arreglar los prefijos mezclados de API	Sprint 8
Corregir posibles fallas con Dockerfile y requirements	Sprint 8
Comprimir imágenes para que no tarden en cargar	Sprint 8
Permitir que empleados puedan cambiar su contraseña	Sprint 8
Transferir clientes a API	Sprint 8
Actualizar servidor a la última versión del repositorio	Sprint 8
Crear componentes reutilizables para todas las vistas	Sprint 8
Unificar estilos de todas las vistas	Sprint 8

- **User story map**

User Story Mapping						
Seguridad	Facturación	Clients	Inventario	Proveedores	Reportes	Diseño / UX
Acceso privado	Generar factura	Correcciones de formularios	Agregar productos	Gestión de proveedores	Reporte últimos 10 días	Mejoras visuales en home
Restringir vistas a usuarios loggeados	Factura por cada transacción	Validar y guardar cliente	Opción de agregar producto	Editar proveedor	Generar reporte con filtros	Unificar estilos en CSS
Cerrar sesión correctamente	Descargar factura en PDF	Guardar cambios de cliente	Modificar cantidad / stock	Guardar proveedor	Visualizar datos completos de pedidos	Modificar logo en vistas
	Autocompletado al editar		Uniformes – método optimizado	Conexión backend-frontend	Resumen de ventas	Corrección de navegación

## Sprint Backlog

- Pila del sprint actual
  - Puntos del sprint

Tarea	Descripción	Puntos de Historia	Tiempo Estimado	Fecha de Entrega
Crear componentes para todas las vistas	separar cada vista en componentes reutilizables para que el código sea más ordenado.	5	3	12/09
Transferir clientes a api	move la lógica de clientes para que se maneje desde la API.	6	2	12/09
Implementar para que los empleados puedan cambiar su contraseña con un código de verificación a su correo	que los empleados puedan cambiar su contraseña con un código que les llegue al correo.	8	2	24/09
Comprimir las imágenes para que no tarden tanto en cargar	reducir el peso de las imágenes para que carguen más rápido.	3	1	12/09
Arreglar problemas con el Docker y sus requerimientos	arreglar los errores de configuración y dependencias en Docker.	5	1	12/09
Arreglar los prefijos mezclados de api	corregir rutas o nombres que están mal combinados en la API.	3	2	12/09
Hacer limpieza de imports (quitar duplicados)	quitar importaciones duplicadas o que no se usan.	2	1	12/09
Unificar las clases duplicadas de telas	juntar en una sola las clases repetidas relacionadas a telas.	3	2	12/09
Cambiar diseño login	darle un nuevo diseño a la pantalla de login.	3	2	24/09
Unificar las clases de ventas	juntar en una sola las clases repetidas de ventas.	3	1	12/09
Arreglar el modelo de la tabla venta	corregir errores en la estructura de la tabla de ventas.	5	2	12/09

Arreglar el duplicado de dominio de ordenes	eliminar configuraciones duplicadas en órdenes.	5	2	12/09
Arreglar los serializer de productos	corregir cómo se devuelven los datos de productos desde la API.	3	3	12/09
Refactorizar los colores del home pedidos billpage, etc (style)	ordenar y mejorar la paleta de colores en esas vistas.	5	1	12/09
Arreglar el serializer de compra	corregir cómo se manejan los datos de compras en la API.	3	2	12/09
Eliminar tablas innecesarias	borrar tablas de la base de datos que no se usan.	2	1	12/09
Darle funcionalidad al botón de enviar en leanding page	hacer que el botón de enviar en la landing funcione de verdad.	3	2	12/09
unificar el style de las vistas leading page, login, registros, forgot pass, contabilidad	que todas esas vistas tengan un mismo estilo.	5	1	12/09
Unificar style de todas las vistas	que todas las pantallas de la app tengan un diseño consistente.	5	2	12/09
Actualizar servidor y que este en la ultima versión del repositorio con 0 errores	dejar el servidor con la última versión del repo y sin errores.	4	2	24/09
Implementar vista de trabajador	crear la pantalla o sección para trabajadores.	5	3	24/09
Pruebas de carga	probar cómo responde el sistema con muchos usuarios al mismo tiempo.	5	1	24/09
Pruebas de volumen	probar con gran cantidad de datos.	5	1	24/09
Prueba de seguridad	revisar que el sistema no tenga fallos de seguridad.	5	1	24/09
Arreglar el problema con el historial de pedidos	arreglar errores en cómo se muestran o guardan los pedidos anteriores.	5	2	24/09
Implementar pinia en todas las vistas	usar Pinia (gestor de estado) en toda la app	8	2	12/09
Refactorizar algunas vistas con style	ordenar y mejorar los estilos en algunas vistas.	5	1	12/09
total		119	46	
Total efectivo		77	33	

- **Calendario de planificación del sprint**

<b>Descripción de la tarea</b>	<b>Fecha de entrega</b>
darle funcionalidad al botón de enviar en landing page	12 Sept 2025
Landing page, login, register, forgot pass, contabilidad	12 Sept 2025
Clientes, clientes registro, proveedores, reporte de ventas (styles)	12 Sept 2025
inventario, home, pedidos, billpage (styles)	12 Sept 2025
implementar pinia en todas las vistas	12 Sept 2025
unificar styles de todas las vistas	12 Sept 2025
crear componentes para utilizar en todas las vistas	12 Sept 2025
Eliminar tablas innecesarias	12 Sept 2025
Arreglar el serializer de compra	12 Sept 2025
Pruebas de Seguridad	24 Sept 2025
Revisar que problemas hay en el sistema y que se deben refactorizar Backend	06 Sept 2025
Arreglar problemas con el historial de pedidos	24 Sept 2025
Pruebas de volumen	24 Sept 2025
Pruebas de carga	24 Sept 2025
Unificar los serializers de producto	12 Sept 2025
Arreglar la duplicación del dominio de ordenes (problema 13)	12 Sept 2025
Arreglar el modelo de la tabla de ventas	12 Sept 2025
Unificar las clases de ventas	12 Sept 2025
Unificar las clases duplicadas de Telas	12 Sept 2025
Hacer limpieza de imports (quitar duplicados)	12 Sept 2025
Arreglar los prefijos mezclados de Api	12 Sept 2025

## Incremento

- **Enlace del repositorio**  
<https://github.com/Ninaswiftie09/BOF>
- **Actividades que se terminaron completamente**

- Crear componentes para utilizar en todas las vistas
- Implementar vista de trabajador
- Transferir Clientes a API
- Implementar para que los empleados puedan cambiar su contraseña con un código de verificación al correo
- Comprimir las imágenes para que no tarden tanto en cargar
- Arreglar posibles fallos con el dockerfile y el requirements
- Arreglar los prefijos mezclados de API
- Hacer limpieza de imports (quitar duplicados)
- Unificar las clases duplicadas de Telas
- Cambiar diseño Login
- Unificar las clases de ventas
- Arreglar el modelo de la tabla de ventas
- Arreglar la duplicación del dominio de órdenes (problema 13)
- Unificar los serializadores de producto
- Arreglar problemas con el historial de pedidos
- Inventario, home, pedidos, billpage (styles)
- Arreglar el serializador de compra
- Eliminar tablas innecesarias  
Darle funcionalidad al botón de enviar en landing page
- Landing page, login, register, forgot pass, contabilidad

- **Ejecución del presupuesto**

- **Porcentaje gastado hasta el momento**

El costo total aproximado de nuestro proyecto, calculado mediante la técnica de

Puntos de Caso de Uso Ajustados (PCUA), es de \$56,999.00 USD.

Hasta el cierre del Sprint 7 se habían ejecutado \$48,749.18 USD ( $\approx 85.5\%$ ).

Durante el Sprint 8, los gastos estuvieron enfocados en nuevas funcionalidades y mejoras visuales, como el cambio de contraseña con verificación por correo, la compresión de imágenes, la limpieza de imports, la refactorización de estilos y el rediseño del login. También se avanzó en correcciones técnicas en backend relacionadas con serializers y modelos.

Inicialmente, el costo de este sprint se había estimado en \$280 USD (46 horas). Sin embargo, al descontar las horas de las tareas no completadas, el gasto real fue de  $\approx \$201$  USD (33 horas efectivas), lo que eleva el acumulado a \$48,950.05 USD, equivalente al  $\approx 85.9\%$  del presupuesto total.

De esta forma, aún queda disponible aproximadamente un 14.1% del presupuesto ( $\approx \$8,049$  USD), que se destinará a las pruebas de carga, seguridad, validación completa y estabilización en producción.

- **Tiempo consumido del tiempo total estimado**

Durante el Sprint 8, el equipo invirtió 33 horas efectivas, lo que representa un progreso acumulado cercano al 76% del tiempo total previsto para el desarrollo principal del producto.

La distribución de tiempo en este sprint fue la siguiente:

- Desarrollo de funcionalidades nuevas y conexión con backend:  $\approx 50\%$
- Corrección de errores y mejoras visuales:  $\approx 35\%$
- Configuración y despliegue técnico:  $\approx 15\%$

## Resultados del Sprint

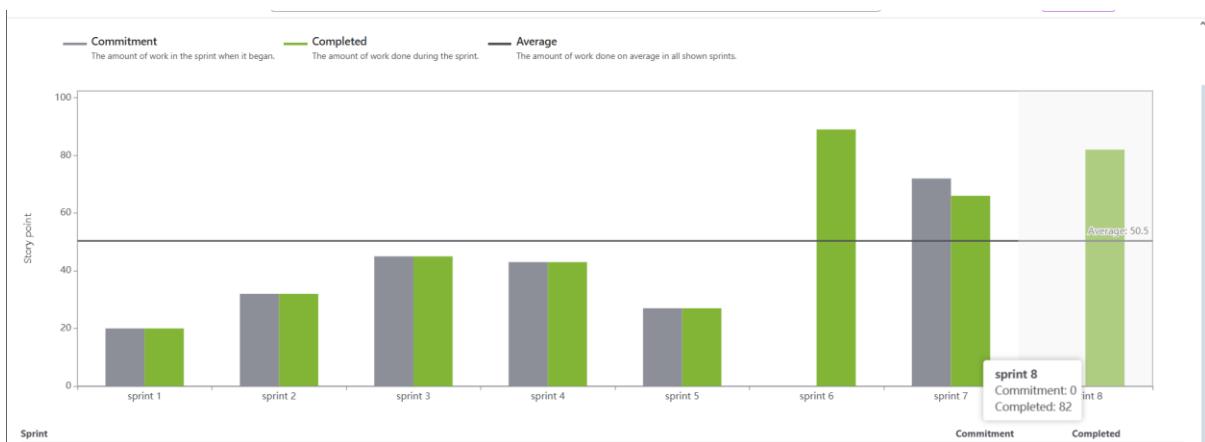
- **Gráfico burndown**

Date - September 7th, 2025 - September 26th, 2025  
Sprint goal - refactorizar backend y frontend para disminuir deuda técnica y problemas en código. implementar nuevas funcionalidades mostrar cambios en el server



- El gráfico burndown muestra la adición de todas las tareas el día 7 de septiembre, debido a que tenemos dos fechas de entrega se presenta una meseta antes del 10 de septiembre, primera fecha de entrega, pero debido a problemas con el programa que no se resolvieron a tiempo, la bajada esperada de la gráfica se presenta días después.

La meseta extensa que sobresale en medio de la gráfica es la semana de descanso, ya se esperaba que apareciera, por ello se planificó la segunda entrega luego de esa semana. El día 24 de septiembre fue la fecha para la segunda entrega, debido a problemas pasados, sucedió lo mismo que en la primera fecha. Por la complejidad que tuvo la refactorización, diversas tareas se atrasaron, algunas que eran base de otras tareas, haciendo un efecto de cascada para los retrasos.



Como se observa en el gráfico de velocidad, en este sprint se completaron 82 de los 119 puntos comprometidos ( $\approx 69\%$  de cumplimiento). Además, la velocidad promedio del equipo aumentó de 44 a 50.5 puntos, impulsada por los sprints recientes; no obstante, persiste una brecha entre lo comprometido y lo entregado que conviene ajustar en la planificación.

- **Discusión del éxito del sprint**

El Sprint 8 fue positivo en términos generales, ya que permitió avanzar en varias funcionalidades y mejoras visibles dentro del sistema, manteniéndolo en un estado estable y operativo. Aunque no se alcanzó el 100% de las tareas comprometidas, lo desarrollado aporta valor inmediato y deja una base sólida para los siguientes ciclos.

Se logró completar aproximadamente el 68.90% de los puntos de historia (82 de 119), destacando avances como la implementación del cambio de contraseña con verificación por correo, la compresión de imágenes para mejorar rendimiento, la limpieza de imports duplicados, la refactorización de estilos en vistas clave y el rediseño de la pantalla de login. También se cerraron correcciones técnicas en el backend relacionadas con serializers y modelos de tablas.

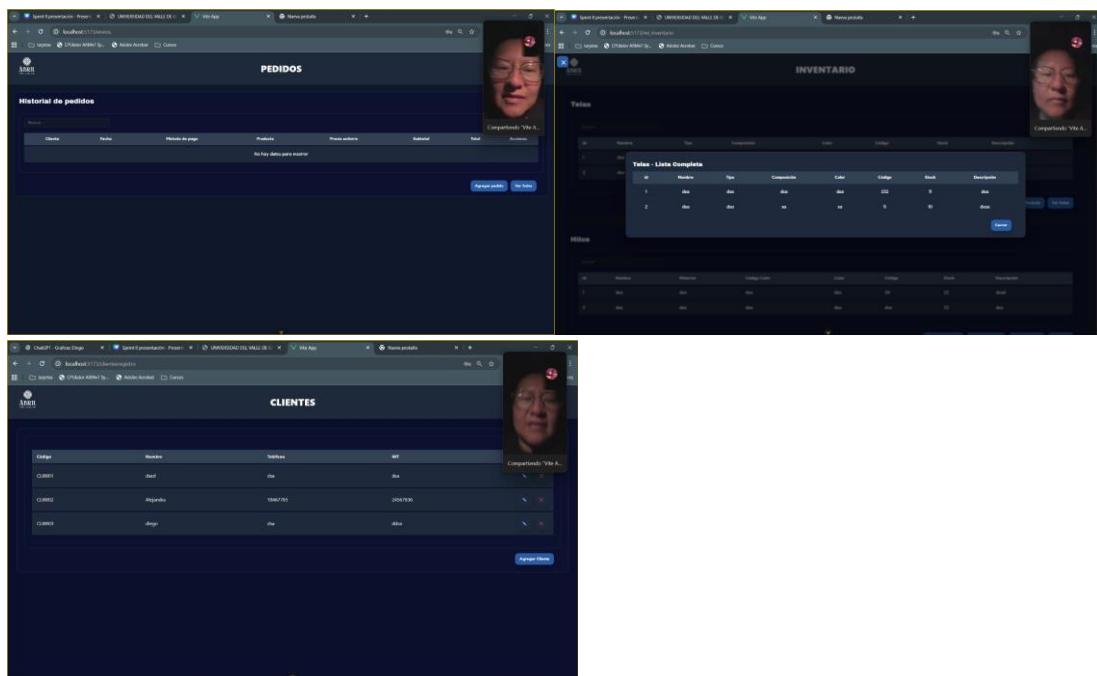
Por otro lado, quedaron pendientes tareas relevantes como la actualización del servidor, la vista de trabajador, pruebas de carga, volumen y seguridad, la integración completa de Pinia en todas las vistas, la corrección del historial de pedidos y algunos refactores de estilo. Estas se trasladarán al Sprint 9.

El gráfico burndown refleja un avance constante, con caídas alineadas a las fechas de entrega (12 y 24 de septiembre). Sin embargo, la concentración de entregas cercanas a la segunda fecha muestra un área de mejora en la distribución del trabajo para evitar acumulación de pendientes.

En cuanto al presupuesto, inicialmente se habían estimado \$280 USD para este sprint (46 horas). Al ajustar y excluir las horas no invertidas en tareas que no se completaron, el gasto real fue de ≈\$201 USD, lo que eleva el acumulado del proyecto a \$48,950.05 USD, equivalente al ≈85.9% del presupuesto total. Este ajuste refleja un uso más realista de los recursos y da mayor margen disponible para las siguientes iteraciones.

En conclusión, el Sprint 8 fue exitoso porque entregó un conjunto funcional y usable de mejoras, mantuvo la estabilidad del sistema y optimizó el uso del presupuesto, aunque con el reto de cerrar tareas críticas pendientes. El foco hacia adelante será equilibrar mejor la carga de trabajo, garantizar el cierre de funcionalidades técnicas claves y fortalecer el bloque de pruebas y validación antes de producción.

- **Evidencias de muestra del incremento desarrollado a usuarios finales**



- **Retrospectiva del sprint**

Durante el Sprint 8 el equipo logró completar aproximadamente el 69% de las tareas comprometidas, lo que permitió entregar avances funcionales como el cambio de contraseña con verificación por correo, la compresión de imágenes, el rediseño del login y la limpieza de imports. El sistema se mantuvo estable y usable para los usuarios finales, con un uso eficiente de los recursos ya que se ejecutó cerca del 72% de las horas planificadas. Sin embargo, se observó una concentración de entregas hacia el final del sprint y quedaron tareas críticas pendientes, lo que evidencia la necesidad de distribuir mejor la carga de trabajo y ajustar la planificación para los siguientes ciclos con el fin de aumentar el cumplimiento.

## Resultados de las pruebas de experiencia de usuario

Los usuarios consideraron positivo el nuevo diseño, destacando que las tablas y la vista en general se perciben claras y agradables. Mencionaron que los colores utilizados les resultan atractivos y que el sistema se ve “bonito”. Como única sugerencia, recomendaron aumentar el tamaño de la letra para mejorar la legibilidad.

## Resultados de las pruebas de carga, estrés y seguridad

- **Prueba de carga**

No se pudieron realizar porque no se completó la refactorización y hay algunas cosas que aún no funcionan.

- **Prueba de estrés**

No se pudieron realizar porque no se completó la refactorización y hay algunas cosas que aún no funcionan.

- **Prueba de seguridad**

No se pudieron realizar porque no se completó la refactorización y hay algunas cosas que aún no funcionan.

## Refactorización

- **Deuda técnica**

Un 40% corresponde a pruebas no ejecutadas de carga, volumen y seguridad, lo que deja en riesgo la validación de estabilidad y rendimiento. Un 25% está asociado a infraestructura y despliegue, principalmente la actualización del servidor. Otro 20% proviene de refactors incompletos, como la integración de Pinia en todas las vistas y la corrección del historial de pedidos. Finalmente, un 15% se relaciona con la estandarización visual y la limpieza de código, donde aún persisten inconsistencias en estilos y duplicados menores que deben resolverse para reducir la complejidad futura y garantizar la calidad técnica del sistema.

- **Áreas que se refactorizaron**

Durante este sprint se realizaron varias mejoras técnicas relevantes: se unificaron las clases duplicadas de ventas y telas, se consolidaron los serializers de productos y de compras, y se corrigió el modelo de la tabla de ventas. Además, se ejecutó una limpieza de imports para eliminar duplicados y referencias innecesarias, se reorganizó la estructura de estilos en vistas clave como inventario, home, pedidos y billpage, y se avanzó en la unificación visual del login y otras pantallas. Estas acciones ayudaron a reducir la complejidad del código, mejorar la mantenibilidad y dar mayor consistencia al sistema en su conjunto.

## Informe de gestión de tiempo

Nina Nájera

Fecha	Inicio	Fin	Tiempo interrumpido (min)	Delta Tiempo (min)	Tarea	Comentarios
12/09	-	-	-	45	Eliminar las tablas innecesarias de la base de datos	-
12/09	-	-	-	45	Unificar los serializers de producto	-
12/09	-	-	-	20	Arreglar la tabla de ventas	-
12/09	-	-	-	30	Unificar las clases de ventas	-
12/09	-	-	-	20	Unificar las clases duplicadas de telas	-
12/09	-	-	-	15	Arreglas los prefijos duplicados en Api	-

Mishell Ciprian

Fecha	Inicio	Fin	Tiempo interrumpido (min)	Delta Tiempo (min)	Tarea	Comentarios
07/09				45	Asignación de puntos de historia para algunas tareas	
12/07				240	Investigación para refactorización	

14/07				480	Arreglo de fetch para todas las vistas	
16/07				120	Botón para enviar correo	No se instalaba bien la librería
18/07				30	Botón para ocultar y mostrar contraseña mientras el usuario la ingresa	

María José Yee

Fecha	Inicio	Fin	Tiempo interrumpido (min)	Delta Tiempo (min)	Tarea	Comentarios
7/09	9:00pm	10:00pm	0	60	Se comenzó a analizar el código y cambios	Se comentó en líneas importantes
8/09	2:00pm	4:30pm	60	90	Analisis del código	Se comentó en líneas importantes
9/09	1:10pm	5:00 pm	60	170	Analisis del código	Se comentó en líneas importantes
10/09	2:50 pm	9:00 pm	60	310	Recopilación de componentes similares para generalizar	Se hizo un resumen y compilación de los componentes
11/09	5:20 pm	11:00 pm	60	280	Recopilación de componentes similares para generalizar	Igual durante todas las vacaciones, un poco cada día

25/09	1:00 pm	10:00 pm	60	480	Merge de Tablas.vue con style borrador	Se intentó juntar el diseño de botones, la nueva estructura, los diseños generales del programa del style borrador con Tablas.vue
26/09	7:00 am	6:50 pm	80	630	Finalizar el style completo de todas las pantallas, componentes, estructura, estilo...	Se almorzó y luego siguió el merge

Diego Ramírez

Fecha	Inicio	Fin	Tiempo interrumpido (min)	Delta Tiempo (min)	Tarea	Comentarios
3/09	1:00 pm	3:00 pm	0	120	Cambiar carpeta Api clientes	
4/09	10:00 am	12:00 pm	0	120	Unificar tablas duplicadas en el orm	
5/09	4:00 pm	5:30 pm	30		Unificar serializers	break

Alejandro Rivera

Fecha	Inicio	Fin	Tiempo interrumpido (min)	Delta Tiempo (min)	Tarea	Comentarios
2025-09-08*	09:00	11:00	10	110	Analizar backend en busca de errores	Semana antepasada – revisión general

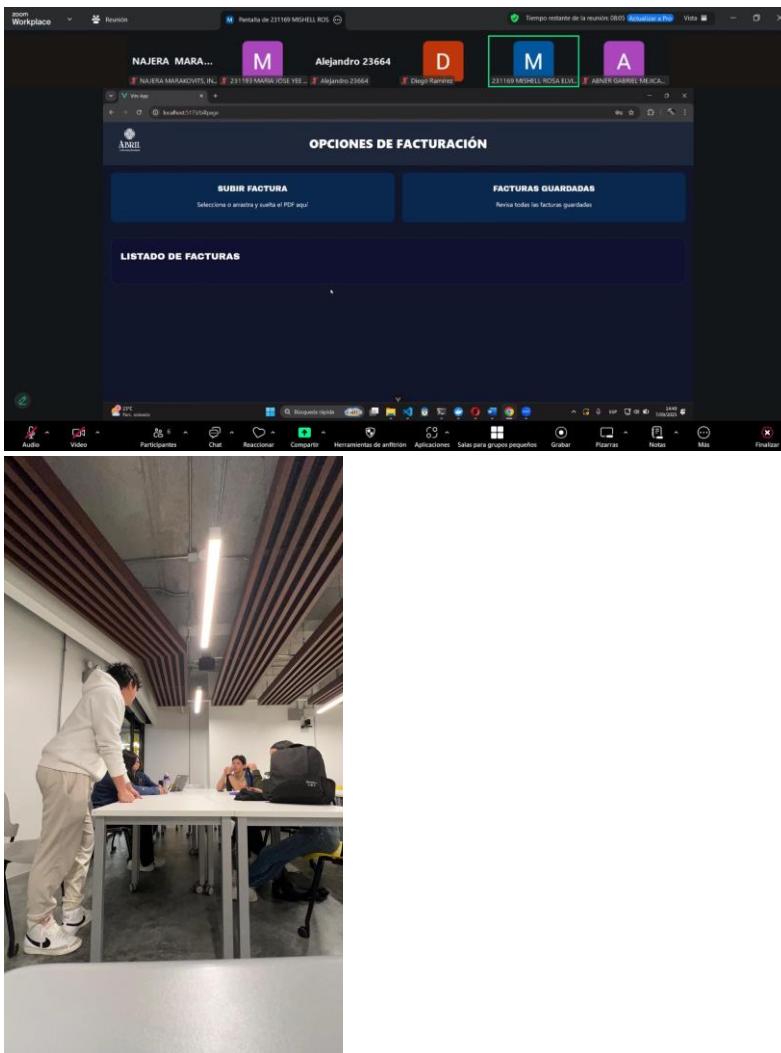
2025-09-15*	14:00	16:00	5	115	Corregir errores asignados	Semana pasada – solo los míos
2025-09-23*	19:00	21:00	0	120	Implementar bloqueos por rol de empleados	Esta semana – integración en el router

Abner Mejicanos

Fecha	Inicio	Fin	Tiempo interrumpido (min)	Delta Tiempo (min)	Tarea	Comentarios
18/09	3:30pm	5:40	40	120	Investigar como factorizar la tabla para que funcione en todas las vistas	El tiempo perdido fue viendo tik tok
19/09	4:00	7:20	20	180	Creando la factorizacion de las tablas	El tiempo perdido fue porque me puse a jugar con mi perro
21/09	8:40	1:00	70	190	Continuacion de la factorizacion de las tablas y retoques en el style para funcionamiento de las tablas y formulario	El tiempo perdido fue desayunando
23/09	4:00	6:30	0	2:30	Se termino la refactorizacion y la union del style con las vistas que me tocaba editar, todas las tablas del programa funcionales	No hay tiempo perdido

## Anexos

- **Reuniones**



- **Evidencia del uso de Jira**

