**Resumen ejecutivo del proyecto *TELTEC SYSTEM***

1. **Alcance - descripcion del problema**

El presente proyecto TELTEC SYSTEM surge como respuesta a la necesidad de la empresa TELTEC NET de optimizar el tiempo y la precisión en sus procesos administrativos, los cuales se realizaban de forma manual, generando ineficiencia, errores en la recaudación y dificultades en la gestión de clientes. Para solucionar este problema, se desarrolló una aplicación web con módulos funcionales como login, gestión de usuarios y clientes, recaudación, gastos, reportes y notificaciones, aplicando la metodología ágil SCRUM. La solución permitió automatizar completamente la administración de clientes y el control de pagos, reducir tiempos de generación de reportes, mejorar la comunicación con usuarios y garantizar la seguridad y disponibilidad de la información, contribuyendo así al fortalecimiento digital de la empresa.

1. **Diseño de la aplicación - diagrama de arquitectura**

**TELTEC SYSTEM** sigue **la arquitectura Modelo-Vista-Controlador (MVC)** **adaptada a una aplicación web moderna con Next.js App Router y API Routes**.

Aplicacion del patrón **MVC**:

* 1. **Modelo (Model)**

Ubicado en: lib/models/\*.ts  
Estos archivos definen la estructura de datos que Prisma mapea hacia la base de datos PostgreSQL.

**Componentes:**

* User.ts
* Cliente.ts
* Pago.ts
* Gasto.ts

**2.2 Vista (View)**

Ubicado en: app/\*\*/\*.tsx  
Cada archivo .tsx en carpetas como /usuarios, /clientes, /gastos, etc., representa una página o componente visual del sistema.

**Componentes:**

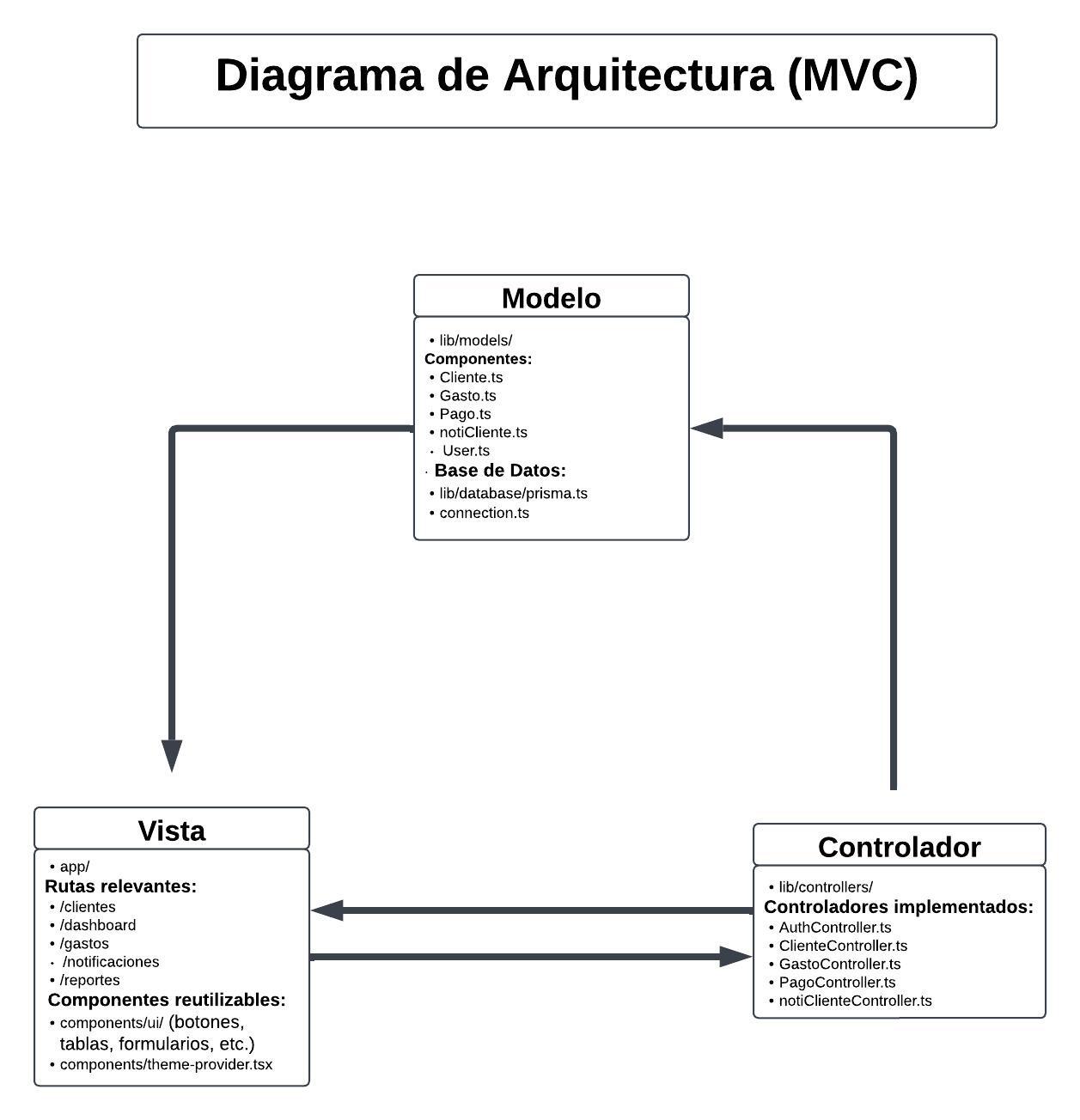
* app/recaudacion/page.tsx
* app/reportes/pagina principal de reportes.tsx
* Componentes UI reutilizables en components/ui/\*.tsx

**2.3 Controlador (Controller)**

Ubicado en: lib/controllers/\*.ts  
Los controladores manejan la lógica de negocio, reciben peticiones desde las rutas /app/api/\*, consultan los modelos y devuelven respuestas.

**Componentes:**

* ClienteController.ts
* PagoController.ts
* GastoController.ts

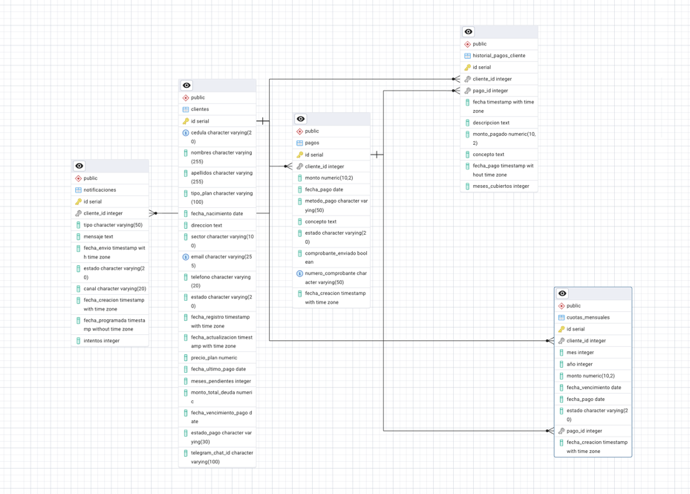


1. **Diseño de la base de datos**

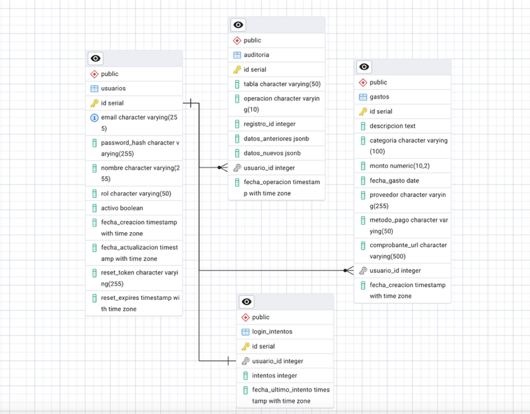
La base de datos de **TELTEC SYSTEM** está diseñada en PostgreSQL bajo un modelo relacional, que garantiza integridad y trazabilidad. Incluye tablas como usuarios, clientes, pagos, gastos, cuotas y notificaciones, relacionadas mediante claves foráneas. Esta estructura permite gestionar de forma eficiente los procesos de recaudación, control de clientes y comunicación, asegurando seguridad, escalabilidad y soporte para reportes e integraciones futuras. Se modelan de forma normalizada, lo que permite un control preciso de los procesos de recaudación y comunicación, y soporta la trazabilidad financiera requerida por TELTEC NET.

**Se utilizó una base de datos relacional**

* **Integridad referencial**: Garantiza la consistencia entre clientes y sus pagos, cuotas, notificaciones, etc.
* **Consultas complejas**: Permite hacer JOINs para obtener reportes por cliente, mes, estado, etc.
* **Escalabilidad vertical adecuada**: Suficiente para empresas medianas como TELTEC NET.
* **Estandarización**: SQL es ampliamente soportado, mantenido y documentado.
* **Seguridad**: PostgreSQL permite roles, permisos y control transaccional robusto.

**Para la administracion en la empresa**

**Para la administracion delaq sistema por parte de los usuarios**

****

**Ventajas del modelo actual**

* Permite auditar todos los pagos y notificaciones por cliente.
* Controla cuotas por mes y compara con pagos reales.
* Puede generar reportes de deuda, mora, pagos, cumplimiento, etc.
* La separación entre cuotas y pagos permite flexibilidad para pagos adelantados, pagos parciales o retrasos.
* Las notificaciones están asociadas a clientes, permitiendo su historial y control de envío.

1. **Planificación - product backlog**

La planificación del proyecto **TELTEC SYSTEM** se llevó a cabo aplicando la metodología ágil **Scrum**, dividiendo el desarrollo en **11 sprints**, cada uno con una duración de **una semana**. Esta estrategia permitió organizar y distribuir eficientemente las tareas técnicas (HT) e historias de usuario (HU) de acuerdo con las prioridades del sistema y las necesidades de la empresa TELTEC NET.

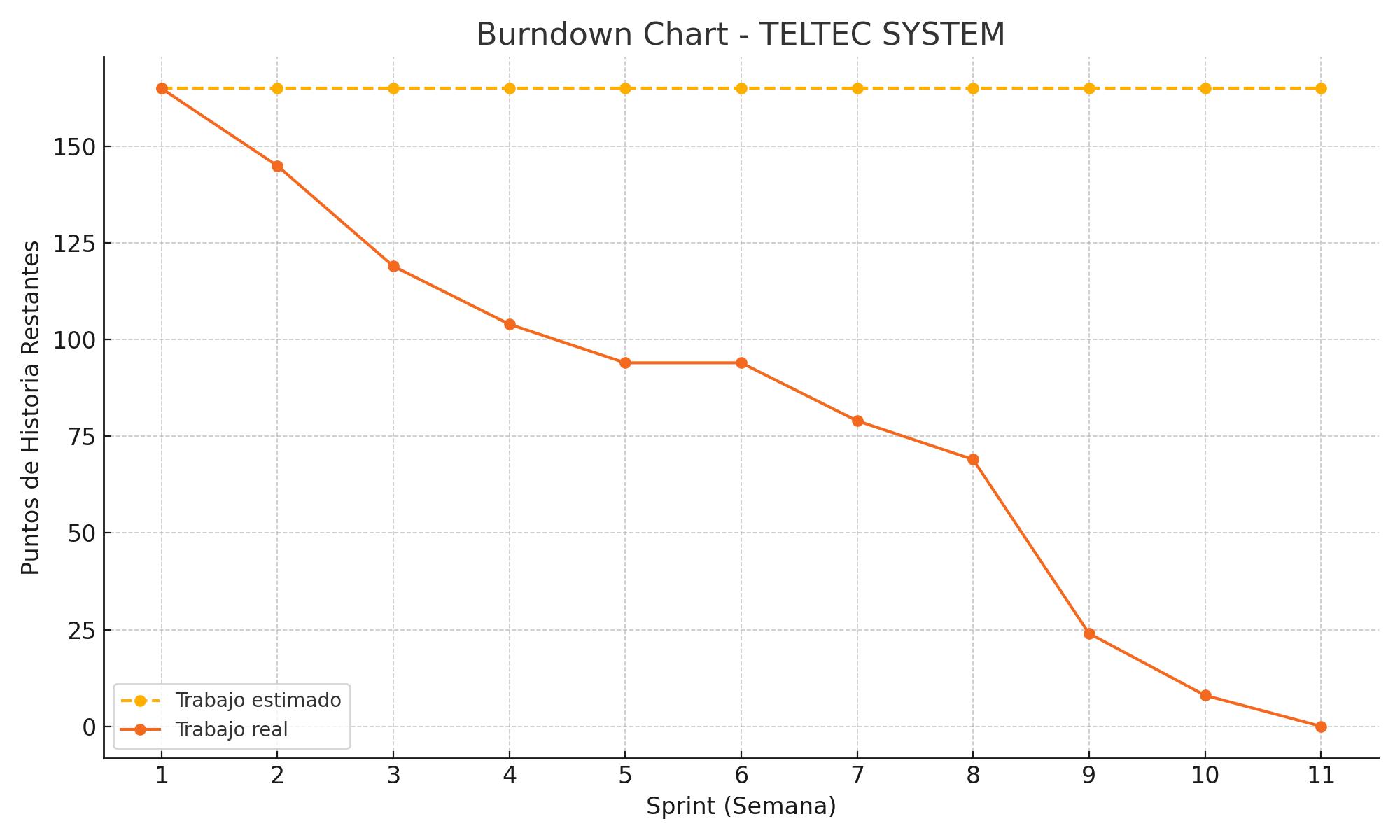
El **Product Backlog** fue elaborado progresivamente e integrado a lo largo del ciclo de desarrollo. Cada sprint estuvo compuesto por funcionalidades concretas, permitiendo una validación continua con el usuario final. Las tareas se estimaron en puntos de historia, facilitando la gestión de tiempo y esfuerzo del equipo.

La planificación contempló los siguientes aspectos:

* **Sprints 1 al 10**: Desarrollo funcional y técnico del sistema, incluyendo autenticación, gestión de usuarios, clientes, recaudación, gastos, reportes y notificaciones.
* **Sprint 11**: Dedicado exclusivamente a la **presentación y defensa final del proyecto**, así como a la documentación, revisión final de funcionalidades, y ajustes sugeridos.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprint** | **Código** | **Descripción** | **Criterio de Aceptación** | **Estimación** | **Estado** |
| **Sprint 1** | HT-001 | Configuración del Entorno de Desarrollo | El entorno debe permitir ejecutar el proyecto con frontend y backend integrados. | 5 | Finalizado |
|  | HT-002 | Diseño de la Base de Datos Relacional | La base de datos debe estar normalizada y contener las relaciones necesarias. | 5 | Finalizado |
|  | HT-003 | Implementación de Autenticación JWT | El sistema debe emitir y validar tokens JWT correctamente. | 5 | Finalizado |
|  | HU-001 | Inicio de Sesión | El usuario puede ingresar sus credenciales para iniciar sesión. | 5 | Finalizado |
|  | HU-002 | Recuperación de Contraseña(falta) | El sistema permite recuperar la contraseña mediante correo. | 5 | pendiente |
| **Sprint 2** | HU-007 | Control de Acceso por Rol | Acceso al sistema restringido por rol: Admin, Cliente, etc. | 5 | Finalizado |
|  | HT-007 | Middleware de permisos | El middleware debe restringir accesos según el rol asignado. | 5 | Finalizado |
|  | HT-002-01 | Deuda Técnica: Implementación de Ruta de los estilos | La hoja de estilos debe estar cargada y aplicada correctamente en todas las vistas. | 3 | Finalizado |
|  | HT-003-01 | Deuda Técnica: Gestión de Usuarios | La gestión de usuarios debe incluir crear, editar, eliminar y listar usuarios. | 5 | Finalizado |
|  | HT-004-01 | Deuda Técnica: Gestión de Roles | El sistema debe permitir asignar y modificar roles a los usuarios. | 3 | Finalizado |
|  | HU-002 | Recuperación de Contraseña | El sistema permite recuperar la contraseña mediante correo. | 5 | Finalizado |
| **Sprint 3** | HT-004-01 | Deuda Técnica: Registro de Cliente, Validación de Cédula y Planes | El sistema debe permitir asignar y modificar roles a los usuarios. | 5 | Finalizado |
|  | HU-003 | Gestión de Clientes | Permite crear, editar y eliminar clientes con validación. | 5 | Finalizado |
|  | HT-004 | Configuración de Roles y Permisos | El sistema debe validar que cada ruta esté protegida por roles. | 5 | Finalizado |
|  | HT-003-01 | Deuda Técnica: Implementación de Login por Roles | La gestión de usuarios debe incluir crear, editar, eliminar y listar usuarios. | 5 | Finalizado |
| **Sprint 4** | HT-004-02 | Deuda Técnica: Búsqueda, Eliminación y Edición de Clientes | El sistema debe permitir buscar, editar y eliminar clientes desde la interfaz. | 5 | Finalizado |
|  | HT-010 | Responsive Design y mejoras UX | La interfaz debe adaptarse correctamente a diferentes tamaños de pantalla. | 5 | Pendiente |
|  | HT-011 | Mejora de carga de formularios | Los formularios deben precargar datos y validar campos antes del envío. | 5 | Pendiente |
| **Sprint 6** | HT-004-02 | Visualización de Recaudaciones | El sistema debe permitir buscar, editar y eliminar clientes desde la interfaz. | 5 | Finalizado |
|  | HU-006 | Generación de Reportes | Reportes financieros y clientes exportables | 5 | Finalizado |
|  | HT-006 | Implementar exportación PDF | Los reportes deben poder exportarse en CSV con formato correcto. | 5 | Finalizado |
| **Sprint 7** | HT-008 | Test unitarios con Jest | Las funciones deben tener cobertura del 80% mediante pruebas con Jest. | 5 | Finalizado |
|  | HT-009 | Cifrado de contraseñas y validación | Las contraseñas deben cifrarse con bcrypt y validarse correctamente. | 5 | Pendiente |
| **Sprint 8** | HU-004 | Gestión de Gastos | Registrar, visualizar y editar gastos con validación | 8 | Finalizado |
|  | HU-005 | Visualización de Recaudaciones por Cliente | Tabla con montos filtrable y ordenable | 5 | Finalizado |
|  | HU-006 | Impresión de Reporte de Recaudación | Reportes financieros y clientes exportables | 5 | Finalizado |
|  | HU-008 | Registro de Pagos Mensuales | Ingresar pagos con fecha, monto y cliente | 5 | Finalizado |
|  | HU-009 | Historial de Recaudación por Cliente | Lista detallada de pagos anteriores | 5 | Finalizado |
|  | HT-016 | Implementación del Módulo de Recaudación | Gestión de pagos, cálculo y saldos | 8 | Finalizado |
|  | HT-014 | Implementación del Módulo de Recaudación | Integración e inserción a base de datos | 8 | Finalizado |
| **Sprint 9** | HU-006 | Generación de Reportes | Reportes financieros y clientes exportables | 5 | Finalizado |
|  | HT-006 | Implementar Exportación CSV | Los reportes deben poder exportarse en CSV con formato correcto. | 5 | Finalizado |
|  | HU-004 | Gestión de Gastos | Registrar, visualizar y editar gastos con validación | 5 | Finalizado |
| **Sprint 10** | HU-006 | Deuda Técnica Generación de Reportes | Reportes financieros y clientes exportables | 3 | Finalizado |
|  | HT-006 | Implementar Exportación CSV | Los reportes deben poder exportarse en CSV con formato correcto. | 2 | Finalizado |
|  | HT-018 | Módulo Notificaciones WhatsApp | El sistema debe registrar el estado de las notificaciones y permitir envío desde la interfaz. | 8 | Finalizado |
| **Sprint 11** | HU-006 | DT: Generación de Reportes - Especificar en el Gráfico | Reportes financieros y clientes exportables | 3 | Finalizado |
|  | HT-006 | Implementar Exportación CSV | Los reportes deben poder exportarse en CSV con formato correcto. | 2 | Finalizado |
|  |  | Defenza del proyecto | Se procedio a la **presentación y defensa final del proyecto**, así como a la documentación, revisión final de funcionalidades. | 20 minutos | Finalizado |

1. **Burndown Chart – Ejecución del Proyecto**



La siguiente gráfica representa el Burndown Chart del proyecto TELTEC SYSTEM, donde se visualiza la evolución del trabajo restante (en puntos de historia) a lo largo de los 11 sprints semanales definidos en la planificación.

En el eje X se representan los sprints (del 1 al 11), y en el eje Y los puntos de historia restantes por completar. La línea discontinua indica el ritmo de trabajo ideal estimado, mientras que la línea continua representa el avance real del proyecto.

Durante el desarrollo, se observaron avances constantes en la mayoría de sprints, con entregas clave durante los Sprints 2, 6 y 8, donde se completaron módulos funcionales como autenticación, recaudación y reportes. El Sprint 5 no presentó avances significativos por ajustes en la planificación interna. Finalmente, en el Sprint 10 se completó la totalidad de funcionalidades, dejando el Sprint 11 exclusivamente para la presentación, revisión y defensa del proyecto.

Este gráfico evidencia una ejecución efectiva, cumpliendo con la mayoría de entregables previstos, y mostrando una disminución constante del trabajo pendiente, lo cual refleja una correcta aplicación de la metodología ágil SCRUM.