



TecniQuest

***Proyect del Taller: Creación de Aplicaciones con
Google Antigravity***

Creado por EDUTECPR
Con el apoyo de inteligencia artificial

© 2025 | Uso exclusivo para propósitos educativos y distribución digital con fines
profesionales

<https://linktr.ee/edutecpr>

Table of Contents

Introducción General al Proyecto TecniQuest	3
Objetivo General del Proyecto	4
Objetivos Específicos del Proyecto	5
Alcance del Sistema (Scope)	5
Arquitectura del Sistema	6
5.1. Estructura General	6
5.2. Componentes Principales.....	6
Descripción Detallada de los Módulos	7
6.1. Panel de Control (Dashboard).....	7
6.2. Módulo de Clientes	7
Integración adicional:.....	8
6.3. Módulo de Equipos.....	8
Funciones adicionales:.....	8
6.4. Módulo de Reparaciones.....	8
Funciones adicionales:.....	9
6.5. Módulo de Inventario	9
6.7. Módulo de Agenda	10
6.8. Módulo de Reportes	10
6.9. Módulo de Administrador.....	11
Base de Datos – Estructura Propuesta (SQLite)	11
Tablas principales:.....	11
Flujo General de Uso del Sistema	12
Prompt para Antigravity (Formato C.R.A.F.T.)	13
R – Rol	13
A – Acción	13
F – Formato	13
Entregables del Taller	14
Conclusión	14

Introducción General al Proyecto TecniQuest

El proyecto *TecniQuest* surge como la necesidad de desarrollar una aplicación completa, funcional y moderna que permita a un taller de reparación de computadoras manejar de forma organizada todos sus procesos diarios. A lo largo de esta conversación se fueron definiendo, ajustando y expandiendo los requisitos del sistema, comenzando con una estructura básica y evolucionando hasta un ecosistema robusto que integra módulos, bases de datos, controles de acceso, manejo de imágenes, facturación automática y personalización visual.

Desde el inicio, se estableció que *TecniQuest* debía contar con un panel de control accesible, un menú lateral con secciones clave —como clientes, equipos, reparaciones, inventario, facturación, agenda, reportes y administración—, y una base de datos SQLite administrada por una interfaz desarrollada en Python. A través de los múltiples *User Input*, se identificaron errores, se solicitaron mejoras y se añadieron nuevas funcionalidades que enriquecieron el alcance del proyecto.

Entre las funciones más destacadas se incluyen la capacidad de almacenar y editar datos relacionados, generar e imprimir facturas, manejar una galería de imágenes con thumbnails y comentarios, y garantizar que las fotos se visualicen tanto en el módulo de equipos como en el de reparaciones. También se solicitó mejorar la experiencia del usuario añadiendo un sistema de login, creación y asignación de roles, controles de acceso dinámicos, visibilidad

condicional de pestañas según los permisos y la posibilidad de restablecer contraseñas.

Finalmente, se expresaron necesidades de accesibilidad y personalización, como integrar un modo oscuro (*dark mode*) y asegurar que todos los botones e interfaces funcionen adecuadamente bajo cualquier tema visual.

En resumen, esta recopilación de requerimientos representa una evolución natural de la aplicación hacia un sistema completo, profesional, flexible y centrado en la eficiencia operativa, pensado para atender tanto las necesidades del administrador como las de los técnicos y personal que lo utilizarán en el taller. Esta introducción contextualiza el conjunto de solicitudes y decisiones que dieron forma al diseño final del proyecto *TecniQuest*.

Objetivo General del Proyecto

El proyecto TecniQuest tiene como objetivo desarrollar una aplicación local, completa, accesible y organizada que permita a un taller de reparación de computadoras administrar sus procesos diarios de forma eficiente. El sistema integrará módulos funcionales conectados a una base de datos, permitiendo registrar clientes, equipos, reparaciones, inventario, facturación, calendario de trabajo, reportes y configuraciones administrativas.

Objetivos Específicos del Proyecto

1. Crear una interfaz moderna, responsive y accesible que permita al usuario manejar todos los módulos sin dificultad.
2. Implementar un panel de control que muestre las actividades del día y métricas importantes.
3. Desarrollar un menú lateral con navegación clara y organizada.
4. Diseñar e implementar una base de datos SQLite que se comunique eficientemente con la interfaz.
5. Integrar controles de acceso basados en roles y permisos de usuarios.
6. Implementar funciones de galería, manejo de imágenes y facturación imprimible.
7. Garantizar que la aplicación sea escalable, editable y funcional en ambiente local.

Alcance del Sistema (Scope)

TecniQuest cubrirá todos los procesos esenciales de un taller de reparación, incluyendo:

- Registro y búsqueda rápida de clientes.
- Administración de equipos asociados a cada cliente.
- Registro detallado de reparaciones, piezas, costos y fotos del progreso.
- Registro y control de inventario.
- Generación automática de facturas.
- Agenda y calendario de citas.
- Módulo de reportes generales y específicos.
- Módulo de administración con ajustes avanzados y control de permisos.

Arquitectura del Sistema

5.1. Estructura General

- **Frontend / UI:** Generado mediante Google Antigravity usando prompts estructurados.
- **Backend Local:** Python (Flask o FastAPI sugerido para quienes lo quieran expandir luego).
- **Base de Datos:** SQLite (archivo local .db).
- **Almacenamiento de Imágenes:** Carpeta local con rutas asociadas en la base de datos.
- **Plantillas HTML:** Integradas con Jinja2 (modelos generados por Antigravity).

5.2. Componentes Principales

1. Interfaz gráfica accesible y responsiva.
2. Módulos conectados a base de datos.
3. Sistema de autenticación y roles.
4. Módulo de facturación imprimible.
5. Manejo integrado de imágenes.
6. Modo claro y modo oscuro.

Descripción Detallada de los Módulos

6.1. Panel de Control (Dashboard)

Debe incluir:

- Resumen del día:
 - Total de clientes nuevos
 - Reparaciones en progreso
 - Reparaciones completadas
 - Inventario crítico
 - Citas del día
- Botones rápidos:
 - Crear cliente
 - Crear equipo
 - Crear reparación
- Indicadores visuales tipo tarjetas (cards).
- Compatible con modo claro y oscuro.

6.2. Módulo de Clientes

Debe permitir:

- Crear, ver, editar y eliminar clientes.
- Datos mínimos:
 - Nombre completo
 - Teléfono
 - Email
 - Dirección
 - Notas
- Historial de servicios asociado.

- Búsqueda rápida por:
 - Nombre
 - Teléfono
 - Fecha
 - Equipo asociado

Integración adicional:

- Ver equipos del cliente directamente desde su perfil.
- Ver reparaciones asociadas.

6.3. Módulo de Equipos

Datos requeridos:

- Tipo de equipo (Laptop, Desktop, Mac, Consola, etc.)
- Marca y modelo
- Número de serie
- Estado general
- Descripción del problema
- Cliente asociado (FK)
- Fecha de entrada
- Fecha estimada de entrega

Funciones adicionales:

- Galería de imágenes:
 - Subir fotos
 - Thumbnails automáticos
 - Comentarios por imagen
- Ver fotos también desde el módulo de reparaciones.

6.4. Módulo de Reparaciones

Debe incluir:

- Equipo asociado
- Cliente asociado
- Descripción del problema
- Diagnóstico
- Piezas utilizadas
- Mano de obra
- Costos y totales
- Estado del trabajo (pendiente, en progreso, finalizado, entregado)

Funciones adicionales:

- Subir imágenes de progreso o evidencia.
- Imprimir informe de reparación.
- Ver fotos del equipo.

6.5. Módulo de Inventario

Debe permitir:

- Registrar piezas, componentes y herramientas.
- Control de cantidades.
- Estado de disponibilidad.
- Notificación cuando existan piezas en mínimo.
- Integración con reparaciones:
 - Descuento automático de piezas al asignarlas.

6.6. Módulo de Facturación

Funciones obligatorias:

- Generar factura PDF.

- Datos:
 - Cliente
 - Detalle de servicios
 - Piezas usadas
 - Subtotal
 - Impuestos
 - Total
- Facturas editables antes de imprimir.
- Histórico de facturas.

6.7. Módulo de Agenda

Debe incluir:

- Calendario mensual y semanal.
- Crear citas.
- Recordatorios.
- Vincular cita con cliente o reparación.

6.8. Módulo de Reportes

Reportes sugeridos:

- Clientes atendidos por fecha.
- Reparaciones en progreso vs completadas.
- Ingresos generados.
- Equipos más reparados.
- Piezas más utilizadas.
- Inventario por agotarse.

6.9. Módulo de Administrador

Incluye:

- Sistema de login
- Creación de usuarios
- Roles y permisos:
 - Técnico
 - Supervisor
 - Administrador
- Control de visibilidad del menú:
 - Tabs ocultos según permisos
- Restablecimiento de contraseñas
- Cambio de modo claro y modo oscuro
- Configuraciones generales del sistema

Base de Datos – Estructura Propuesta (SQLite)

Tablas principales:

- clientes
- equipos
- reparaciones
- inventario
- facturas
- usuarios
- roles
- permisos
- imagenes
- agenda
- reportes_generados

Incluye en cada tabla:

- ID autonumérica

- Campos requeridos
- Llaves foráneas
- Fechas de creación y modificación

Accesibilidad y Experiencia del Usuario

El sistema debe cumplir con buenas prácticas:

- Contraste adecuado
- Fuentes legibles
- Navegación clara
- Botones grandes y accesibles
- Compatible con modo claro y oscuro
- Alertas visuales y auditivas
- Validación de formularios

Flujo General de Uso del Sistema

1. Usuario inicia sesión.
2. Accede al Dashboard.
3. Crea o busca un cliente.
4. Registra o selecciona un equipo.
5. Inicia una reparación.
6. Selecciona piezas del inventario.
7. Sube fotos del proceso.
8. Genera la factura.
9. Guarda registro para reportes.
10. Finaliza o entrega el equipo.

Prompt para Antigravity (Formato C.R.A.F.T.)

C — Contexto

Estoy desarrollando una aplicación llamada **TecniQuest**, diseñada para talleres de reparación de computadoras. Necesito que construyas la interfaz completa, moderna, accesible y responsiva con menú lateral y un dashboard funcional.

R — Rol

Actúa como un experto en diseño UI/UX y desarrollo de aplicaciones responsivas con estándares modernos.

A — Acción

Genera una interfaz limpia, profesional y accesible que incluya:

- Dashboard
- Menú lateral
- Módulos (Clientes, Equipos, Reparaciones, Inventario, Facturación, Agenda, Reportes, Administración)
- Formularios
- Tablas
- Modal de login
- Compatibilidad con dark mode

F — Formato

Entrégalo como un proyecto completo en HTML, CSS y JavaScript compatible con Antigravity, con componentes listos para integrar con backend Python.

T – Tono

Profesional, claro, moderno y enfocado en usabilidad.

Entregables del Taller

Los participantes deberán entregar:

1. Proyecto generado en Antigravity con los módulos definidos.
2. Estructura HTML/CSS/JS organizada.
3. Documento del proyecto (este mismo).
4. Evidencia de funcionamiento del menú y el dashboard.
5. Capturas de pantalla del sistema funcionando.
6. Explicación breve del diseño implementado.

Conclusión

TecniQuest representa un proyecto completo, profesional y perfectamente adecuado para enseñar a los estudiantes cómo transformar un concepto funcional en una aplicación visual usando Google Antigravity. Además, deja la puerta abierta para futuras integraciones con backend en Python, ampliación de la base de datos y evoluciones del sistema.