Nombre del proyecto: Taller Mecánico / Mechanic-Kings

Integrantes (**Roles**):

Manuel Collao (**Desarrollador Full-Stack - DBA**),

Yzan Baez (**Scrum Máster y Desarrollador Full-Stack**),

Benjamin Vidal (**Product Owner y Qa Tester**),

Anibal Perez (**Front end - Qa tester**)

-Problemática a resolver

* Crear un Sistema para el Registro y Control del servicio de mantenimiento en taller mecánico, el cual pueda registrar los datos del cliente e ir administrando las mantenciones del vehículo en base el número telefónico del cliente.

-Este proyecto tiene una **complejidad media-alta** debido a los siguientes factores:

* **Múltiples Módulos**: Incluye desarrollo de una página web responsive, gestión de stock, agenda de horarios, y un chatbot, cada uno con funcionalidades específicas y complejas.
* **Seguridad y Disponibilidad**: Requiere garantizar seguridad de datos y disponibilidad 24/7, lo que añade complejidad en términos de arquitectura y desarrollo.
* **Integración y Flujo de Trabajo**: El proyecto requiere una integración eficiente entre todos los módulos, asegurando que trabajan en conjunto de manera fluida.
* **Escalabilidad y Mantenimiento**: La necesidad de escalar y mantener el sistema a lo largo del tiempo también incrementa la complejidad.

-Objetivos Generales

* Lograr agendar citas para mantenimiento/revisión de vehículo del cliente.
* Venta de productos Automotriz.

-Objetivos específicos

* Registrar datos tanto de los clientes como de su vehículo.
* Llevar un control de los clientes registrados
* Llevar un registro de mantenimientos de los vehículos
* Administrar las mantenciones del vehículo.
* Administrar el stock de productos

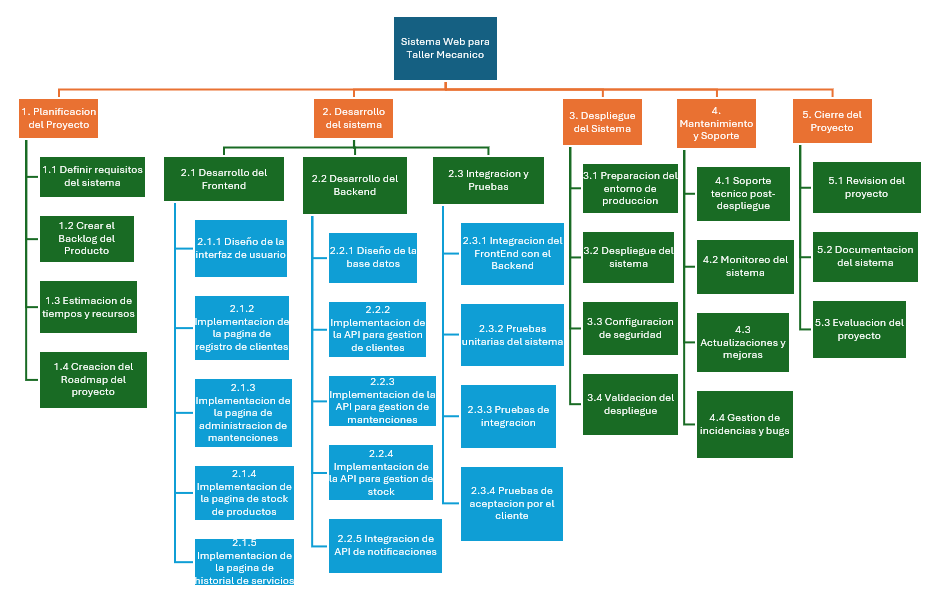
-Metodología a ocupar:

* Metodología Ágil Scrum

-Por qué Scrum:

* Scrum es ideal para proyectos con múltiples áreas como el tuyo porque ofrece adaptabilidad ante cambios, permite entregas incrementales de funcionalidades completas y prioriza la retroalimentación constante. Facilita la organización mediante sprints, asegurando un enfoque en las necesidades del usuario final y una priorización clara de tareas críticas como la seguridad y disponibilidad del sistema. Además, permite al equipo ajustarse rápidamente a nuevas demandas o requerimientos, garantizando un desarrollo más eficiente y alineado con las expectativas del cliente.

Diagrama EDT (dividido por sprint)



-Arquitectura a usar

* Arquitectura Monolítica Modularizada

-Requerimientos

-Página web responsive:

* Intuitiva para el usuario
* Login y Registro de usuarios
* Catálogo de productos disponibles
* Productos asociados a sus respectivas categorías
* Chatbot para la ayuda de los clientes

-Sistema de Gestión de stock

* Baterías
* Aceite de kilometraje
* Neumáticos
* Lubricantes
* Anticongelantes
* Iluminación (ampolletas)
* Accesorios

-Agenda de horarios

* Agendar horarios disponibles durante el mes
* Agendar chequeos
* Agendar citas para mantenimiento

-Guía de cambios

* Servicios:
  + Recambio de aceite
  + Scanner
  + Revisión del vehículo
  + Cambio de disco de embrague
  + Desabolladura.
  + Pintura.
  + **Mecánica** general.
  + Mantenimiento preventivo.
  + **Mecánica** de elevado grado apoyado por tecnologías de punta.
  + Importación y comercialización de repuestos legítimos.
* Productos:
  + Productos que se muestran en sistema de Gestión de stock
* Chatbot:
  + Solicitud de algún problema
  + Disponibilidad de horas
  + Catálogo de los productos
  + Agendar Citas en el Celular
* Reportes y análisis
  + Generar reportes sobre los servicios realizados
* Seguridad
  + Encriptación de contraseña
  + Autenticación de usuario
  + Garantizar la seguridad de los datos
* Disponibilidad
  + El sistema debe estar disponible de Lunes a Viernes
* Tecnologías Utilizadas
  + Lenguajes de programación (HTML, CSS, JavaScript, etc.).
  + Frameworks y librerías (Django y Twilio).
  + Bases de datos (Oracle SQL Developer).
  + Herramientas de desarrollo (VSCode, GitHub).