



UNIVERSIDAD TECNOLÓGICA
NACIONAL
FACULTAD REGIONAL DE VILLA
MARÍA

Entrega N°3

ASIGNATURA: Programación Avanzada
DOCENTE: Vanzetti, Juan José

INTEGRANTES:
Botta, Francisco
Bortolin, Stéfano
Falco, Augusto
Frighetto, Franco
Lopez Bruno, Valentino

Plan de Proyecto	3
Introducción	3
Visión General del Proyecto	3
Entregable del Proyecto	3
Material de Referencia	4
Organización del Proyecto	4
Responsabilidades	4
Proceso de Gestión	5
Objetivos y Prioridades de Gestión	5
Supuestos, dependencias y restricciones	5
Gestión de Riesgos	5
Proceso Técnico	5
Métodos, herramientas y técnicas	5
Documentación de Software	6
Calidad	9
Plan de pruebas	10
Monitoreo	10

Plan de Proyecto

Introducción

Visión General del Proyecto

El proyecto tiene como objetivo principal el desarrollo y la implementación de un Sistema de Gestión (integral) para el taller de autos "Autos S.A". El mismo permitirá llevar a cabo la gestión de clientes, gestión de órdenes de trabajo, gestión de stock del taller y la generación de informes y estadísticas para la toma de decisiones basadas en datos dentro de la empresa. Todos estos aspectos serán implementados para su utilización en un software informático que se le entregará al cliente en cuestión.

El proyecto se desarrollará a través de una serie de actividades en las que se incluyen:

- Análisis de Requerimientos
- Diseño del sistema
- Implementación del sistema
- Generación de Plan de Proyecto
- Estimación de requerimientos.
- Pruebas de software.
- Revisiones y control de calidad del software.

Entregable del Proyecto

Versión	Descripción	Entregable	Fecha Entrega	Lugar Entrega
1.0	Funcionalidad inicial que permite la gestión de vehículos, marca, modelo, clientes y técnico.	Gestión de vehículos, marca, modelo, clientes y técnicos.	11/09/23	A definir.
1.1	Funcionalidad que permita registrar las órdenes de trabajo	Gestión orden de trabajo	09/10/23	A definir.
1.1.2	Funcionalidad que permite asignarle un estado a las órdenes de trabajo.	Gestión orden de trabajo	21/12/23	A definir.
1.2	Funcionalidad que permite crear un documento con datos, a modo de 'factura'.	Gestión factura	21/12/23	A definir.

Material de Referencia

Título	Informe N°	Autor	Organización Publicada	Referencia
Norme IEEE 1058.1	1	Cobos Lomeli Manuel Alejandro Lopez Rivera Jose Miguel Hernande Hernandez Aaron Prof. Margarita Maria de Lourdes Sanchez	Universidad Autónoma Metropolitana	http://aniei.org.mx/paginas/uam/CursoAI/IEEE_rep.pdf

Organización del Proyecto

La organización encargada del proyecto se limitará a las actividades relacionadas con la comunicación constante con el cliente, la captación de requerimientos y el desarrollo e implementación de los mismos. Por otro lado, llevará a cabo la autoevaluación utilizando como base las retrospectivas dadas por el cliente y de esta forma desempeñará una serie de actividades para la reorientación del proyecto, en caso de que fuese necesario.

Por otra parte, el cliente, limita su participación en el proyecto a las actividades de informe de requerimientos que desea, así como también a la mantención y/o actualización de los mismos en caso de que fuese necesario. A su vez y siempre que lo considere pertinente podrá estar presente en las reuniones llevadas a cabo diariamente por la organización del proyecto para conocer distintos aspectos, como por ejemplo el avance del mismo. Finalmente será el encargado de evaluar los posibles incrementos de producto que la parte desarrolladora del software le presente y realizar una retroalimentación de la misma para poder realizar las mejoras que considere necesarias, en caso de que lo fuese.

Responsabilidades

Actividad	Responsable	Consultado	Informado
Comunicación con el Cliente	Equipo de Analistas	Equipo de Desarrolladores, QA	Equipo de Ingenieros
Captación de Requerimientos	Equipo de Analistas	Equipo de Ingenieros	Equipo de Ingenieros
Desarrollo e Implementación	Equipo de Desarrolladores	Equipo de QA	Equipo de Analistas, Ingenieros
Autoevaluación y Retrospectivas	Equipo de Desarrolladores	Equipo de QA, Analistas	Equipo de Ingenieros
Mantenimiento y actualización	Cliente	Equipo de Analistas	Equipo de QA, Desarrolladores, Ingenieros
Evaluación de Crecimientos	Cliente	Equipo de Desarrolladores	Equipo de QA, Analistas, Ingenieros

Proceso de Gestión

Objetivos y Prioridades de Gestión

Durante el desarrollo del proyecto se tendrán en cuenta los siguientes objetivos y prioridades:

1. **Entregar un sistema de gestión funcional:** Lo primordial es desarrollar un sistema que cumpla con todos los requisitos especificados.
2. **Satisfacción del cliente:** se debe asegurar que el software cumpla con las expectativas y necesidades del cliente.
3. **Calidad del software:** Es muy importante para la gerencia la fiabilidad, el rendimiento, la seguridad y la capacidad de mantenimiento del software.
4. **Cumplimiento de plazos:** Mantenerse dentro del cronograma planificado es crucial para la entrega puntual y para evitar retrasos costosos.
5. **Colaboración del equipo:** Fomentar la colaboración y la comunicación efectiva dentro del equipo de desarrollo para mantener la eficiencia y la moral del equipo.

A medida que transcurra el proyecto y nuestros objetivos y prioridades cambien/evolucionen los iremos identificando en el presente apartado.

Supuestos, dependencias y restricciones

Supuestos:

- El cliente está predispuesto para la comunicación continua.
- Disponibilidad de recursos para iniciar el proyecto.
- Aceptación de cambios.

Dependencias:

- Adquisición de licencias.
- Capacitación de personal.

Restricciones:

- Tiempo y recursos.
- Normativas del país.

Gestión de Riesgos

Este apartado no aplica.

Proceso Técnico

Métodos, herramientas y técnicas

Para el análisis y desarrollo de este proyecto hemos decidido aplicar métodos basados en ideologías ágiles, lo cual nos permitirá llevar a cabo el mismo, haciendo utilización del marco de trabajo definido por "SCRUM". Por otra parte haremos utilización de la notación UML para la representación y comprensión del dominio, específicamente los "Diagramas de Clase" que provee dicha notación, así como cualquier otro que nos sea de ayuda durante el transcurso del proyecto.

Para la implementación haremos utilización del lenguaje de programación “Java” version 20.x, acompañado de distintas tecnologías como:

- **Framework Angular** (frontend)
- **Framework Spring Boot** (back end)
- **PostgreSQL** (base de datos)

Por otra parte, utilizaremos distintas técnicas que nos permita identificar los requerimientos del cliente y que podamos reconocer la complejidad de los mismos para así lograr una correcta organización en las actividades a desarrollar. Las técnicas en cuestión son:

- **Historia de Usuario:** Técnica utilizada para la captura y representación de los requerimientos.
- **Poker Planning:** Técnica para llevar a cabo la estimación de los requerimientos y conocer la complejidad de los mismos.

Documentación de Software

- **Product Backlog:** es un artefacto para la visualización de las necesidades individuales del product owner y donde el equipo Scrum puede saber el alcance del proyecto.

Gestión de cliente

- Registrar cliente
- Consultar cliente
- Programar cita con cliente
- Visualizar calendario
- Enviar confirmaciones de citas

Gestión de órdenes de trabajo

- Crear orden de trabajo
- Asignar técnico
- Consultar órdenes de trabajo
- Consultar estado de orden de trabajo

Gestión de Stock

- Registrar entrada de repuesto
- Registrar salidas del repuesto

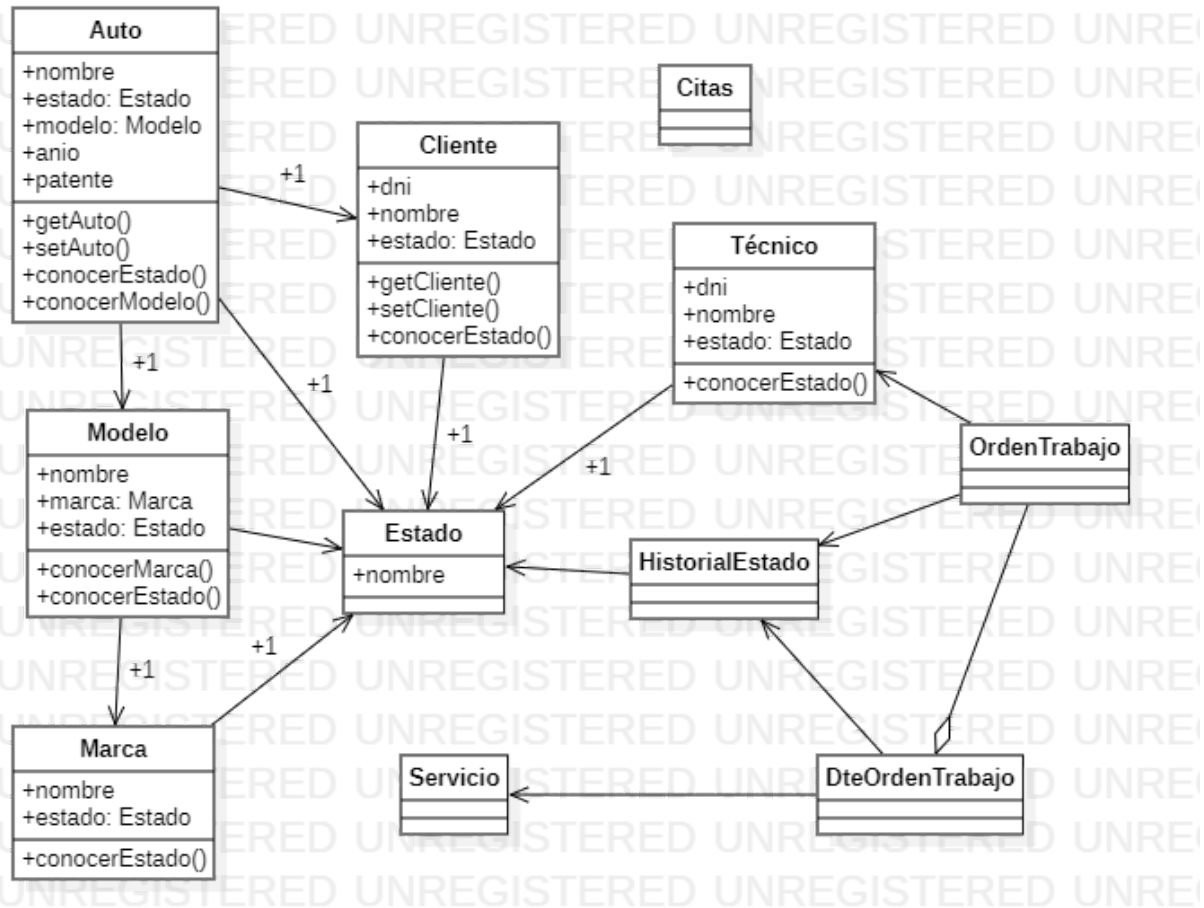
Informes y Estadísticas

- Generar informe de ventas periódicas
- Generar servicios más solicitados

Gestión de vehículos

- Registrar vehículo
- Consultar vehículos
- Registrar marca

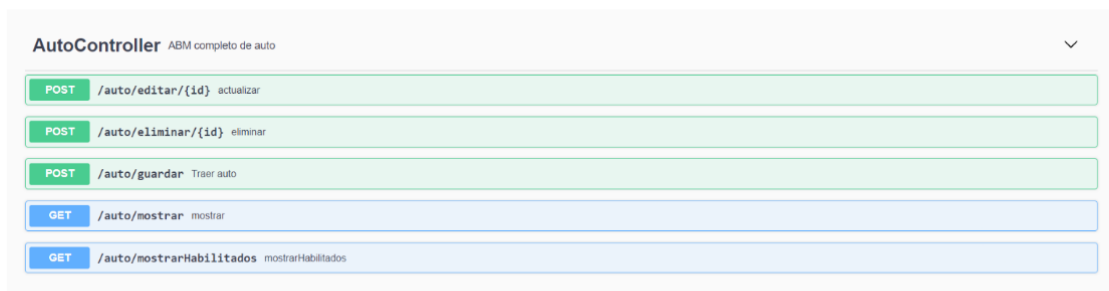
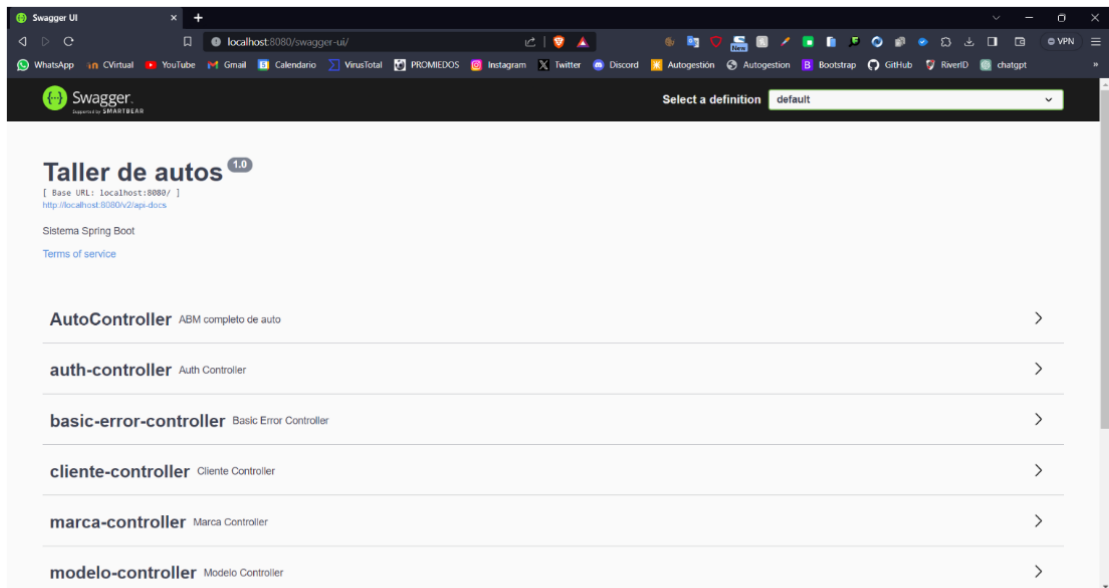
- **Diagrama Entidad-Relación:** es un tipo de diagrama de flujo que ilustra entidades, personas, objetos o conceptos que intervienen en el sistema y se relacionan entre sí. Su uso principal será para el diseño y depuración de bases de datos.



- **Diagrama de Paquetes:** Cada paquete representa una unidad de gestión del product backlog, con las historias de usuario que se tienen en cuenta.



- **Documentación de APIs:** Habíamos implementado [Swagger](#) para realizar la documentación, pero teníamos que bajar la versión del proyecto para que sea compatible, por lo que decidimos no dejarlo, pero sacamos captura de como eran nuestros endpoints.



POST

/auto/editar/{id} actualizar

Try it out

Parameters

Name	Description
<div> <div>id</div> <div>required</div> </div> <div>integer(\$int32)</div> <div>(path)</div>	<div>id</div> <div>id - id</div>
<div> <div>model</div> <div>required</div> </div> <div>object</div> <div>(body)</div>	<div>model</div> <div>Example Value Model</div> <div> <pre>{ "anio": "string", "cliente": { "dni": 0, "estado": true, "id": 0, "nombre": "string" }, "estado": true, "id": 0, "modelo": { "estado": true, "id": 0, "marca": { "estado": true, "id": 0, "nombre": "string" }, "nombre": "string" }, "patente": "string" }</pre> </div> <div> <div>Parameter content type</div> <div>application/json</div> </div>

Responses

Response content type

*/

Code	Description
200	<div>OK</div> <div>Example Value Model</div> <div> <pre>{ "body": {}, "statusCode": "ACCEPTED", "statusCodeValue": 0 }</pre> </div>
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

Ahora usamos [Postman](#), y en la documentación guardamos una colección con algunos [endpoints](#).

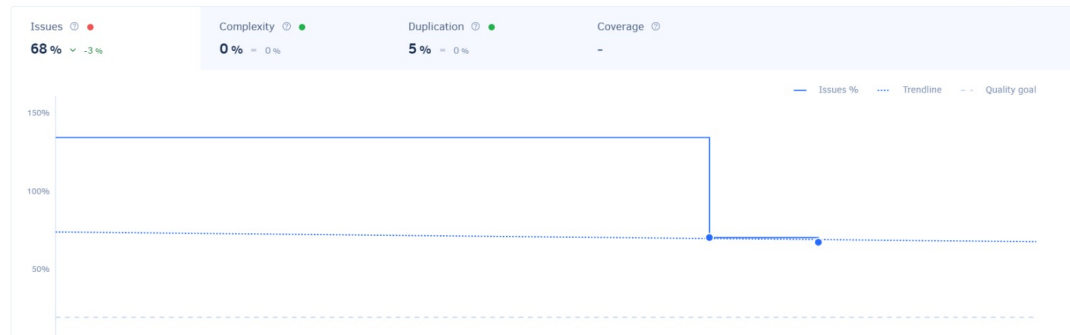
Calidad

Para apoyarnos en este proceso utilizamos la herramienta [Codacy](#). A través de los issues realizamos algunas correcciones:



Quality evolution

Last 3 months Last 31 days Last 7 days



Obteniendo una mejora del 3%, ya que consideramos que las otras correcciones no eran justificadas.

[Commit con las mejoras.](#)

Plan de pruebas

Realizamos un plan detallado de pruebas para el proyecto.

[Plan de pruebas](#)

Monitoreo

Utilizamos la propia herramienta de Java, que viene con JDK llamada JConsole

