Desarrollo del Sistema SUNT	Versión: 2.0
Equipo de trabajo: Tomas Marin, Simón Cárdenas, Juan Pablo Yepes, Juan Andrés Vera	

Entrega 2: Documento de Arquitectura

Contenido

Sección 1: Aspectos generales de la entrega

El propósito de este documento es tener un consenso entre los miembros del equipo de trabajo en el diseño de la arquitectura de software de la aplicación SUNT. La arquitectura planteada en este documento se basa en los siguientes frentes:

- Vista Lógica: visión desde los principales elementos y principios del diseño.
- Vista Física: visión desde la distribución del procesamiento entre los dispositivos.

Sección 2: Evaluación Sprint anterior

En el sprint anterior no tuvimos historias de usuario a desarrollar ya que principalmente se enfocó a la definición del problema , definir las historias de usuario del todo el proyecto y en si a planear y definir todos los temas referentes a el proyecto.

Sección 3: Planificación Sprint actual

Las historias de usuario que vamos a realizar para este Sprint son HU01, HU04, HU06, HU10, HU11, HU18, HU20, entre los cuales se encuentran los 3 registros, de usuario, vehículo y licencia como también algunos de funcionamiento de la página y chat.
Como grupo hacemos reuniones de manera presencial cada semana.

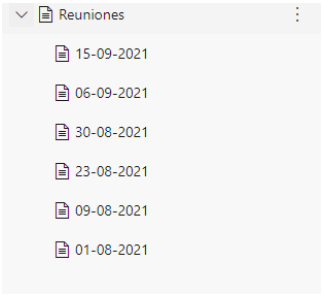
Tabla de historias de usuario aplicando planing poker.

HU01	5.5
HU04	5.5
HU06	7.8
HU10	4.4
HU11	5.8

Comentado [tma1]: [Indique cuáles son las Historias de Usuario que desarrollará para este Sprint, teniendo en cuenta que ellas impactarán la selección de arquitectura y demás artefactos listados en las otras secciones. Incluya elementos como: evidencias de una retrospectiva, evidencias de daily, evidencia de haber aplicado la técnica de poker planning. Refleje los cambios en el Backlog en Azure.]

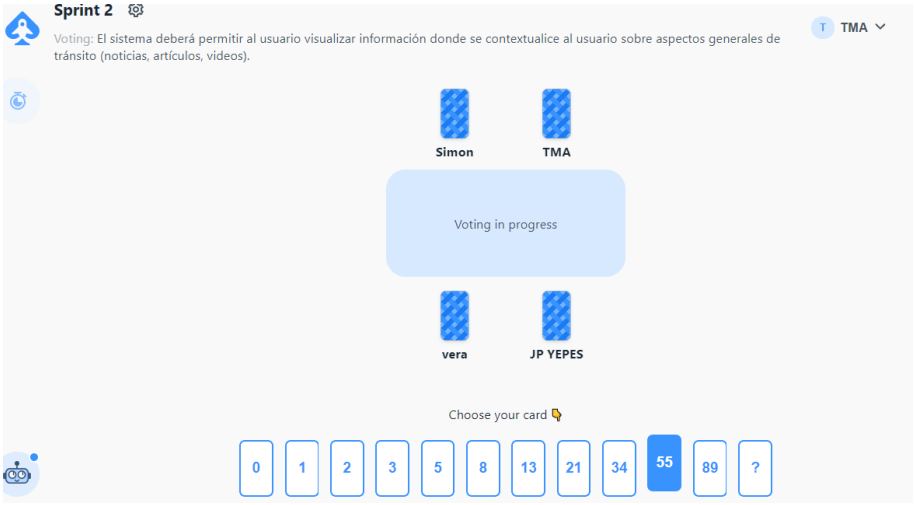
HU18	3.5
HU20	5.3

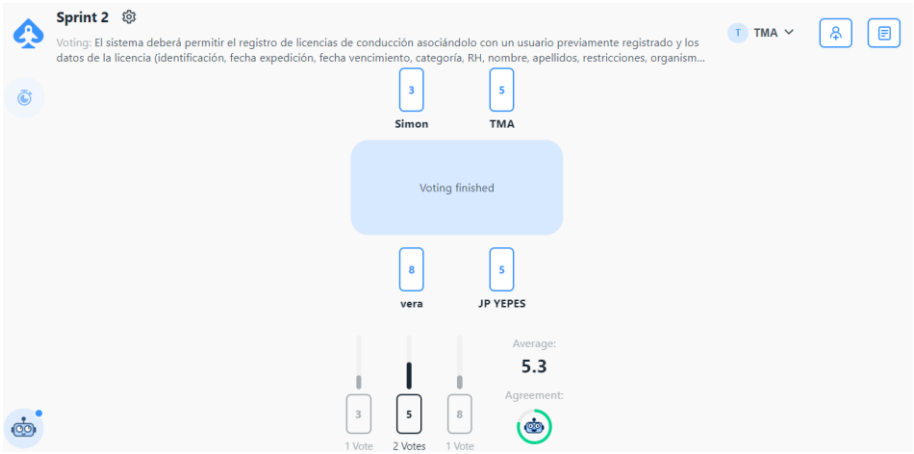
Link reuniones que realizamos de manera semanal :
https://dev.azure.com/tmarina0307/SUNT%20%20PI

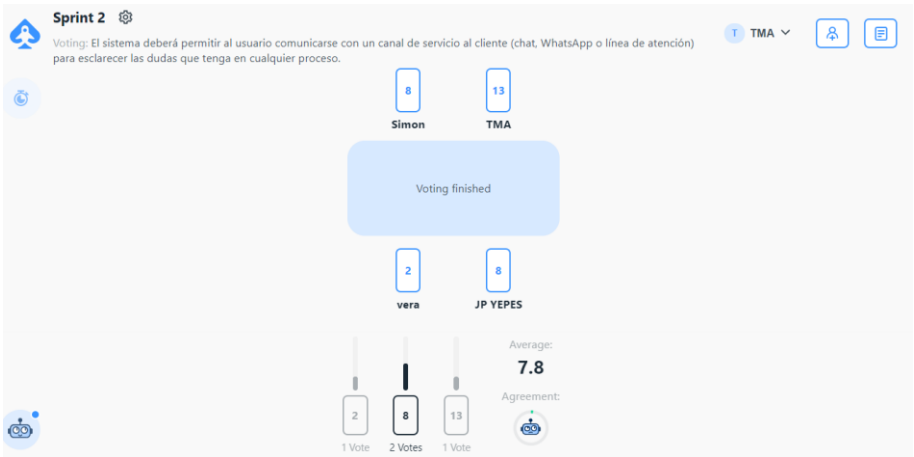


En el link a continuación se encuentra la retrospectiva de la primera entrega del proyecto
https://docs.google.com/document/d/1RAAL7u82Qm5eL0n1c4nNWMOXpWjfX4UfQUWzgTDyvsI/edit?usp=sharing

Evidencia aplicación de técnica de planing poker:




Sprint 2
Voting: El sistema deberá permitir el registro de licencias de conducción asociándolo con un usuario previamente registrado y los datos de la licencia (identificación, fecha expedición, fecha vencimiento, categoría, RH, nombre, apellidos, restricciones, organism...
TMA
3
Simon
5
TMA
Voting finished
8
vera
5
JP YEPES
3
1 Vote
5
2 Votes
8
1 Vote
Average:
5.3
Agreement:


Sprint 2
Voting: El sistema deberá permitir al usuario comunicarse con un canal de servicio al cliente (chat, WhatsApp o línea de atención) para esclarecer las dudas que tenga en cualquier proceso.
TMA
8
Simon
13
TMA
Voting finished
2
vera
8
JP YEPES
2
1 Vote
8
2 Votes
13
1 Vote
Average:
7.8
Agreement:

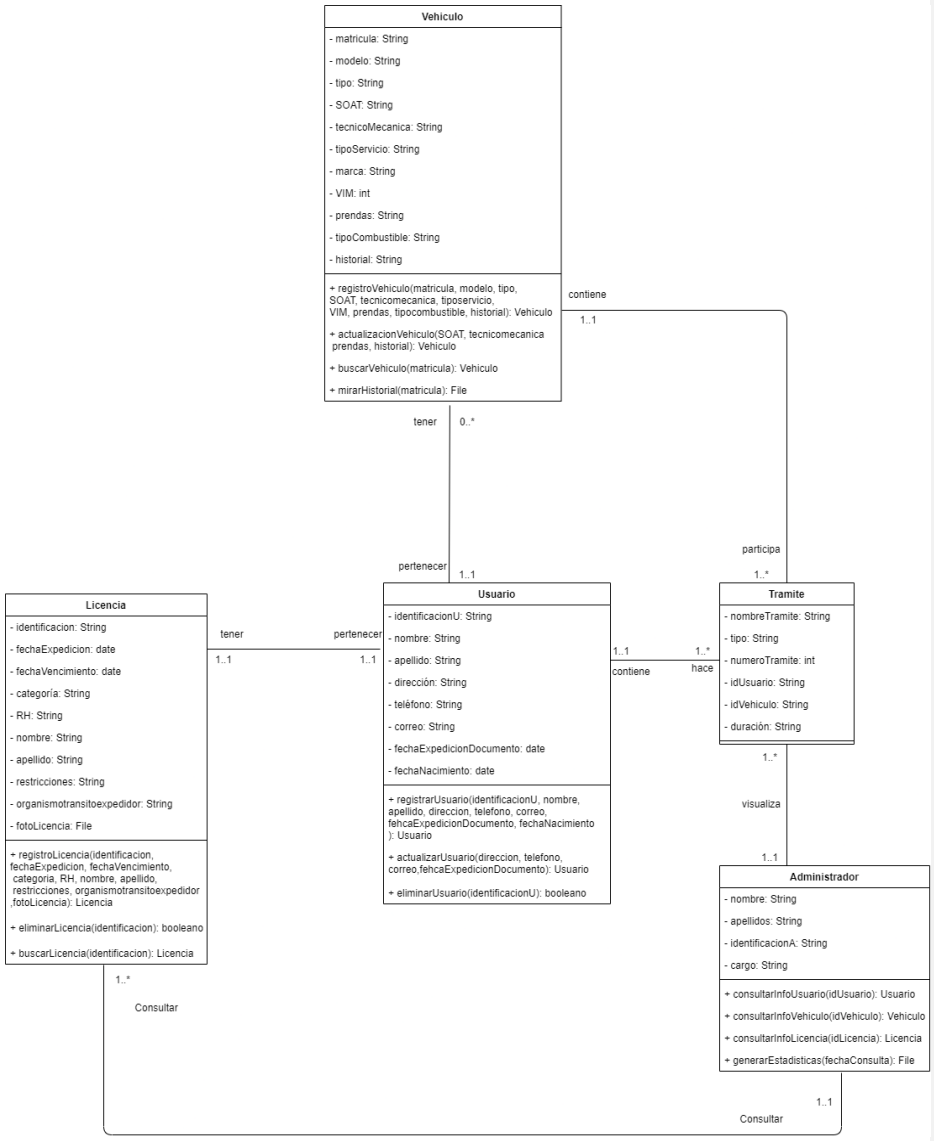
Sección 4: Aspectos estructurales y arquitectónicos de la solución

1.1 Estilos Arquitectónicos Usados

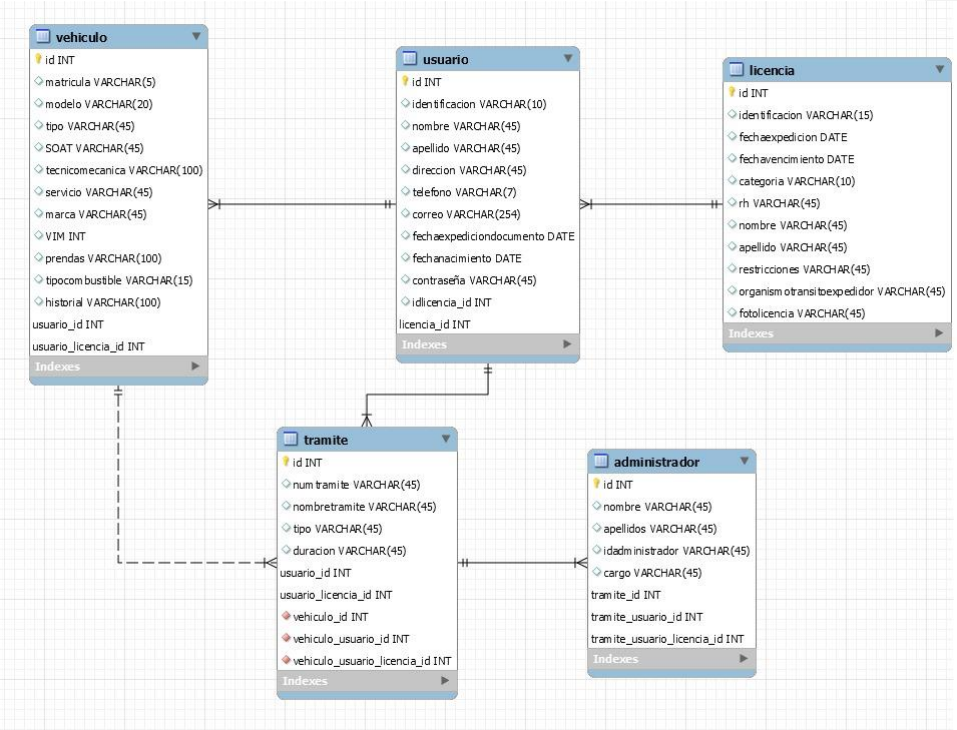
Tipo Aplicación:	Web
Estilo Arquitectónico:	• Client/Server (Implementación): Porque queremos hacer modificaciones fácilmente sin que este todo compacto, como también al tener los recursos centralizados hace que el manejo de datos sea más eficiente y esta arquitectura nos da beneficios en la escalabilidad y encapsulamiento. • Layered (Estructura): Layered porque usamos una arquitectura donde se diferencian las capas de presentación, persistencia y lógica de negocio haciendo más fácil el desarrollo y mantenimiento de la aplicación en el tiempo.
Lenguaje programación	Python
Aspectos técnicos	Usaremos una base de datos relacional que en concreto es SQlite3
Frameworks	Usaremos como framework DJANGO

1.2 Lógica - Diagrama de Clases de Diseño.

Comentado [tma2]: [Explicación: Muestra los componentes principales de diseño y sus relaciones de forma independiente de los detalles técnicos y de cómo la funcionalidad será implementada en la plataforma de ejecución. Los arquitectos crean modelos de diseño de la aplicación, los cuales son vistas lógicas del modelo funcional y que describen la solución]
[Modelo a construir - **Diagramas de Clases de Diseño (refinado):** si se considera que da mayor claridad, se construyen diagramas de clases correspondientes al diseño del sistema, con clases que describen los conceptos de la solución o muestren la forma en la que se aplican los patrones]

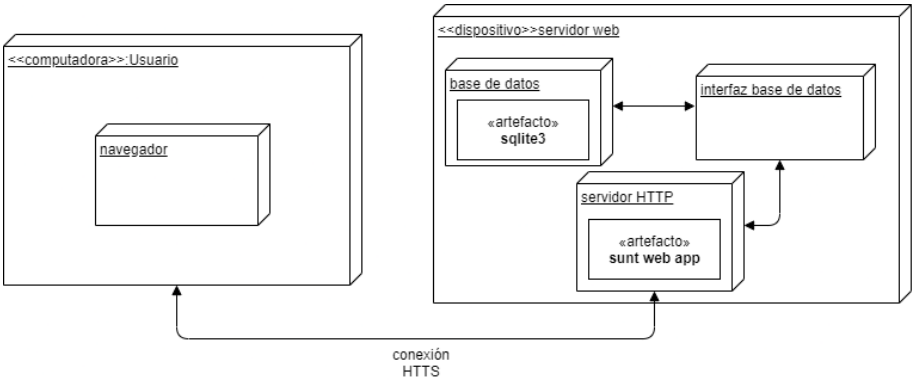

Vehiculo
- matricula: String
- modelo: String
- tipo: String
- SOAT: String
- tecnicoMecanica: String
- tipoServicio: String
- marca: String
- VIM: int
- prendas: String
- tipoCombustible: String
- historial: String
+ registroVehiculo(matricula, modelo, tipo, SOAT, tecnicomecanica, tiposervicio, VIM, prendas, tipocombustible, historial): Vehiculo
+ actualizacionVehiculo(SOAT, tecnicomecanica prendas, historial): Vehiculo
+ buscarVehiculo(matricula): Vehiculo
+ mirarHistorial(matricula): File
contiene
1..1
tener
0..*
pertenecer
1..1
participa
1..*
Licencia
- identificacion: String
- fechaExpedicion: date
- fechaVencimiento: date
- categoria: String
- RH: String
- nombre: String
- apellido: String
- restricciones: String
- organismotransitoexpedidor: String
- fotoLicencia: File
+ registroLicencia(identificacion, fechaExpedicion, fechaVencimiento, categoria, RH, nombre, apellido, restricciones, organismotransitoexpedidor ,fotoLicencia): Licencia
+ eliminarLicencia(identificacion): booleano
+ buscarLicencia(identificacion): Licencia
tener
1..1
pertenecer
1..1
Usuario
- identificacionU: String
- nombre: String
- apellido: String
- dirección: String
- teléfono: String
- correo: String
- fechaExpedicionDocumento: date
- fechaNacimiento: date
+ registrarUsuario(identificacionU, nombre, apellido, direccion, telefono, correo, fehcaExpedicionDocumento, fechaNacimiento): Usuario
+ actualizarUsuario(direccion, telefono, correo,fehcaExpedicionDocumento): Usuario
+ eliminarUsuario(identificacionU): booleano
1..1
contiene
1..*
hace
Tramite
- nombreTramite: String
- tipo: String
- numeroTramite: int
- idUsuario: String
- idVehiculo: String
- duración: String
1..*
visualiza
1..1
Administrador
- nombre: String
- apellidos: String
- identificacionA: String
- cargo: String
+ consultarInfoUsuario(idUsuario): Usuario
+ consultarInfoVehiculo(idVehiculo): Vehiculo
+ consultarInfoLicencia(idLicencia): Licencia
+ generarEstadisticas(fechaConsulta): File
1..*
Consultar
1..1
Consultar

1.3 Vista Lógica - Diagrama Entidad-Relación (Modelo de datos o de persistencia)



Comentado [tma3]: [Explicación: Los diagramas entidad-relación se usan a menudo para diseñar o depurar bases de datos relacionales en los campos de ingeniería de software, sistemas de información empresarial, educación e investigación, para más información: https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion.]
[Modelo a construir - **Diagramas Entidad-Relación:** se recomienda crear este diagrama en una herramienta que permita crear la base de datos a partir de ese modelo, ejemplo: si utilizará una base de datos relacional recomendamos utilizar MySQL workbench, herramienta que permite modelar la base de datos y genera las consultas para la creación de esta.]

1.4 Vista Física - Diagrama de Despliegue.



Comentado [tma4]: [Explicación: En algunas referencias de la literatura esta vista es llamada Vista de Despliegue o Distribución. Ilustra la distribución del procesamiento entre los distintos equipos que conforman la solución, incluyendo los servicios y procesos de base. Los elementos definidos en la vista lógica se "mapean" a componentes de software (servicios, procesos, etc.) o de hardware que definen más precisamente cómo se ejecutará la solución]

[Modelo a construir - **Diagramas de Despliegue:** incluyendo nodos y componentes, e ilustrando la forma en que estos se comunican e interactúan para la distribución del sistema]

1.5 Atributos de Calidad

Comentado [tma5]: [Lista con las consideraciones de los atributos de calidad relevantes (RNF): usabilidad, eficiencia, seguridad, extensibilidad, confiabilidad, portabilidad, mantenimiento, etcétera. Estas características tienen significación especial, deben ser claramente delineadas, **use métricas**]

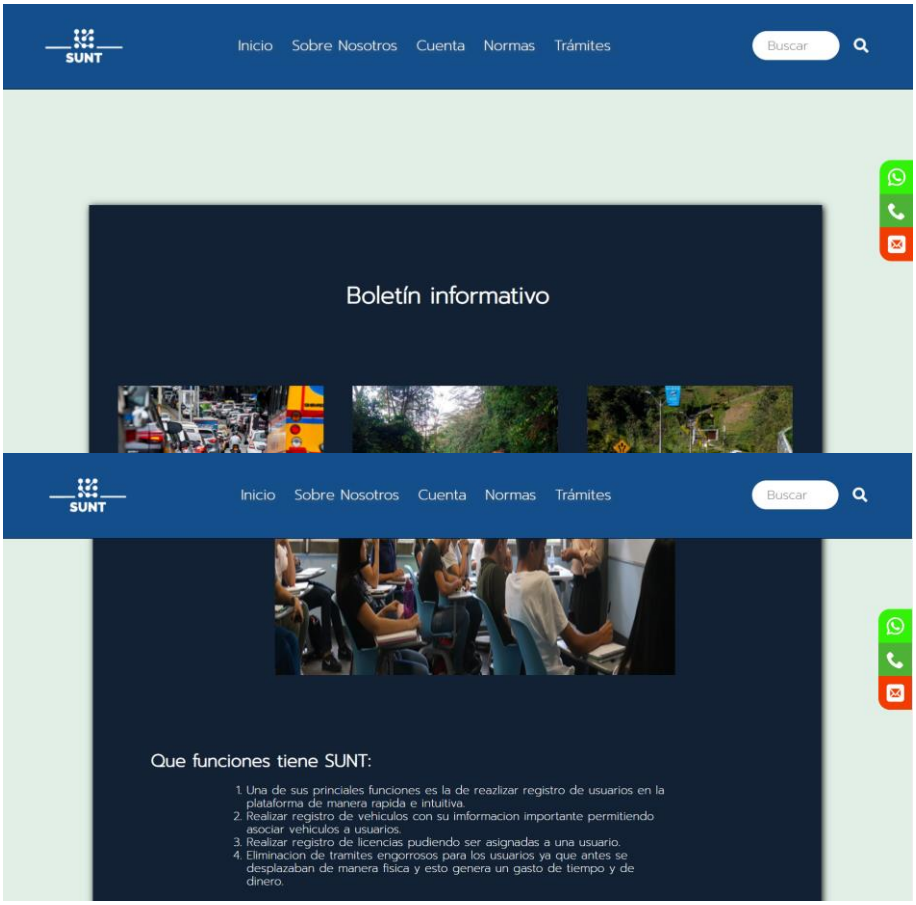
Atributo de Calidad	Descripción	Métrica o método de validación (Para ideas leer la norma ISO/IEC 25023)
Usabilidad	SUNT debe de ser lo suficientemente intuitivo y claro para que los usuarios entiendan las opciones que tiene cada menú.	Para determinar y validar este requisito se realizarán encuestas (preguntas abiertas) sobre el uso de la plataforma donde el usuario podrá evaluar diferentes aspectos de la página web.
Accesible	SUNT debe de mostrar siempre el menú de navegación para que las personas sepan donde se encuentran y puedan moverse con mayor libertad en la página web.	Para determinar y validar este requisito se verificará que el menú siempre sea visible para el usuario respetando las normas de diseño de páginas web http://www.activainternet.es/7-reglas-basicas-buen-diseno-web/ .
confiabilidad	SUNT debe de mostrar la información de tránsito de la página constantemente actualizada y que sea coherente.	Para determinar y validar este requisito se verificará constantemente la información depositada para ver errores o inconsistencias mínimo cada mes.
Fácil navegación	SUNT debe de permitir desplazarse por la página web sin tenerse que moverse por más de 5 ventanas haciendo más fácil su uso.	Para determinar y validar este requisito se probará con la ayuda de varias personas, observando si es fácil de manejar y que si se puede mover por 5 ventanas.
Satisfacción	SUNT debe permitir calificar la satisfacción de los usuarios para así mejorar los problemas que se tengan en la página web.	Para determinar y validar este requisito se realizarán encuestas sobre el uso de la plataforma donde el usuario podrá evaluar diferentes aspectos (de 1 a 5) de la página web y dependiendo de que tan bien los usuarios califiquen un aspecto este se dejará igual o se mejorará.
Inclusión	SUNT debe contar con imágenes descriptivas para que las personas discapacitadas puedan interactuar con la página.	Para determinar y validar este requisito se realizarán pruebas con personas invidentes y se comprobara que las imágenes y demás si proporcionen la información necesaria y adecuada.
Seguridad	SUNT debe de contar con un proceso de validar usuarios	Se verificará que el tiempo de respuesta si sea el adecuado y que a la

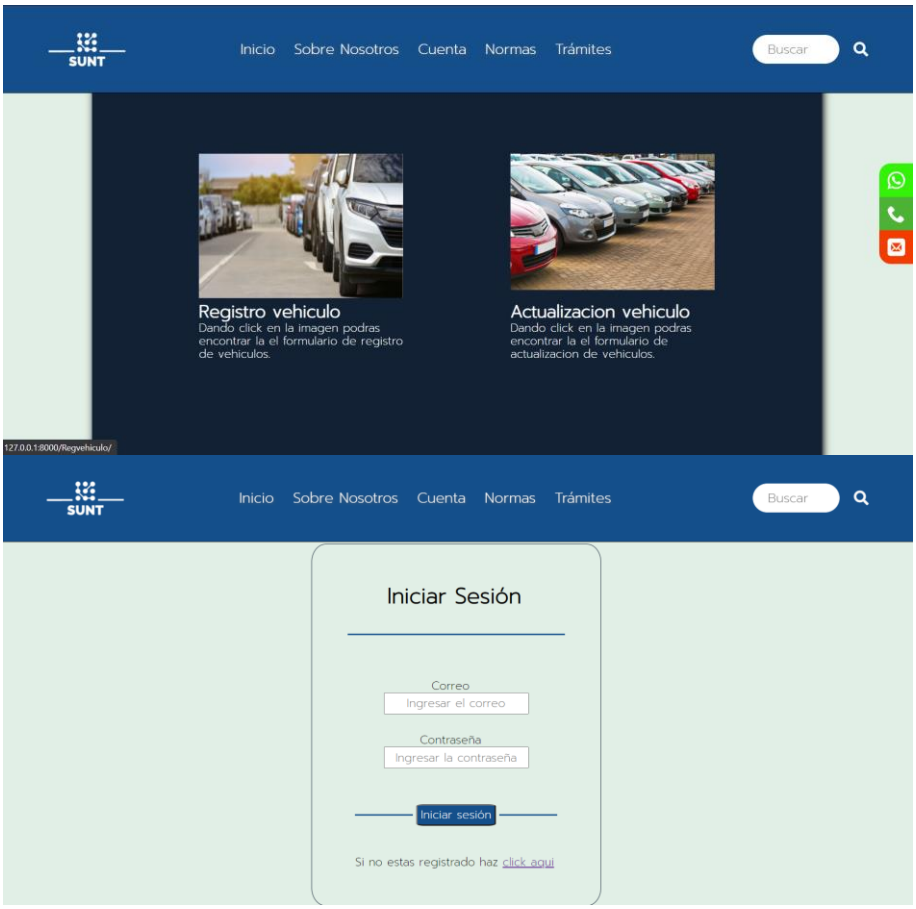
	rápido ósea que no se demore más de 3 segundos en determinar si el usuario esta registrado y debe hacerlo sin mostrar información personal.	hora de hacer una consulta no muestre información privada.

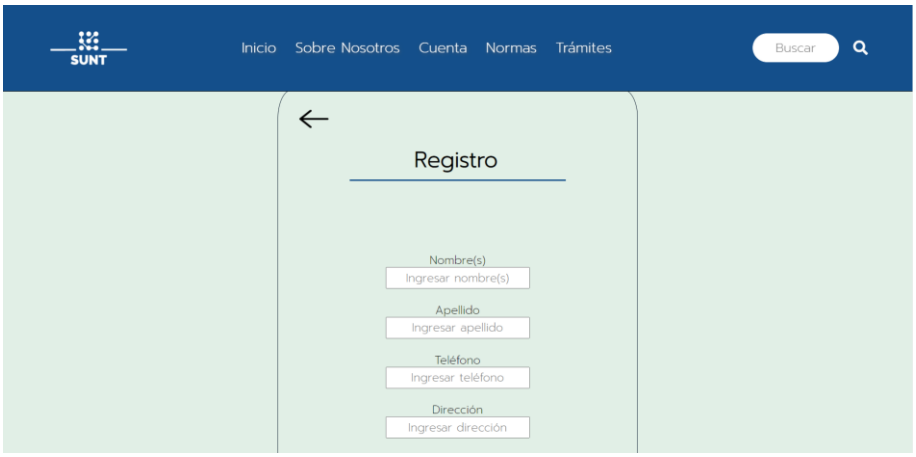
Sección 5: Avances en cuanto a funcionalidad y demostración

Comentado [tma6]: [Incluya fotos con las pantallas ya desarrolladas de la aplicación, y también la URL donde se pueden consultar estas interfaces y la funcionalidad que hay desarrollada hasta el momento. Podría ser un enlace a Github, o una carpeta compartida en algún gestor en la nube]

Link GitHub: https://github.com/tmarina1/PI.git




SUNT
Inicio
Sobre Nosotros
Cuenta
Normas
Trámites
Buscar
Registro vehiculo
Dando click en la imagen podras encontrar la el formulario de registro de vehiculos.
Actualizacion vehiculo
Dando click en la imagen podras encontrar la el formulario de actualizacion de vehiculos.
127.0.0.1:8000/Regvehiculo/
SUNT
Inicio
Sobre Nosotros
Cuenta
Normas
Trámites
Buscar
Iniciar Sesión
Correo
Ingresar el correo
Contraseña
Ingresar la contraseña
Iniciar sesión
Si no estas registrado haz click aqui



Link video (min 8:10 demostración prototipo página): https://eafit-my.sharepoint.com/:v:/g/personal/scardenasv_eafit_edu_co/EQPNcsJTBWREiQSv7z-fWAAB6x6xTYghL_DbDcid8hvinA

Conclusiones y lecciones aprendidas

- Uno de los aspectos que aprendimos en esta iteración fue conocer y entender el funcionamiento de Django.
- Comprendimos como se conectaba la parte de Django y la base de datos con el frontend de la página.
- Entendimos la importancia de la comunicación entre los responsables de la implementación, ya que es vital tener unos objetivos y métodos congruentes en todo el equipo para garantizar que los distintos módulos funcionen entre si correctamente y presenten una solución homogénea.
- Aprendimos que, con este framework, el tratamiento de datos es mucho más fácil y seguro de manejar, respecto a otros frameworks.

Referencias