

Fecha de entrega: 29 de  
septiembre 2024

# PROYECTO FASE II

## QNAVE

Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ciencias y Sistemas  
Análisis y Diseño de Sistemas 2  
Ing. Claudia Liceth Rojas Morales  
Tutor del curso: Brandon Mauricio Noj Romero  
Sección: "A"



Guatemala 01 de septiembre  
Segundo Semestre 2024

# OBJETIVO GENERAL

Objetivo	Indicador
Diseñar una arquitectura de software robusta y escalable para la gestión de transporte privado, que cumpla con los requerimientos técnicos y funcionales especificados, garantizando la integridad estructural y coherencia en la arquitectura de software, para permitir una mayor eficiencia en la ejecución de sus funcionalidades, optimizando el rendimiento y facilitando la adaptabilidad a futuros cambios.	<ol style="list-style-type: none"><li>1. cumplimiento del plazo de entrega.</li><li>2. La participación de todos los integrantes.</li><li>3. cumplimiento de los requisitos especificados en documentación presentada en fase anterior.</li><li>4. Eficiencia del diseño en la ejecución de funcionalidades.</li><li>5. Escalabilidad del diseño arquitectónico</li></ol>

## OBJETIVOS ESPECÍFICOS

Objetivo	Indicador
Realizar la arquitectura del sistema utilizando principios de control de versiones, garantizando una implementación segura y coordinada que permita gestionar de manera eficiente los cambios en la infraestructura, minimizando los riesgos de errores y facilitando la colaboración entre los equipos de desarrollo y operaciones	<ol style="list-style-type: none"><li>1. Aplico correctamente los principios de control de versiones.</li><li>2. Aplicación correcta de cada uno de los ambientes.</li></ol>
Implementar la arquitectura del sistema utilizando estilos arquitectónicos seleccionados, asegurando la configuración y ubicación correctas de los componentes, y aplicando patrones de diseño para garantizar la escalabilidad y la mantenibilidad futura.	<ol style="list-style-type: none"><li>1. componentes del sistema implementados según el estilo arquitectónico seleccionado.</li><li>2. Porcentaje de componentes correctamente configurados y ubicados en la arquitectura</li></ol>
Diseñar todas las funcionalidades y características no funcionales de la arquitectura, asegurando que esta cumpla con los requisitos establecidos y se adhiera estrictamente a los criterios definidos en la documentación técnica, garantizando la alineación con los principios de diseño.	<ol style="list-style-type: none"><li>1. requisitos funcionales y no funcionales cubiertos en el diseño de la arquitectura.</li><li>2. funcionalidades diseñadas que cumplen con los principios de diseño de arquitectura.</li></ol>
Que el estudiante aprenda a utilizar Docker para el despliegue de una arquitectura de software, garantizando la creación de entornos de ejecución consistentes y reproducibles, permitiendo una configuración simplificada, minimizando los problemas de compatibilidad y facilitando la automatización del despliegue y la escalabilidad del sistema en diferentes entornos.	<ol style="list-style-type: none"><li>1. Número de contenedores creados y utilizados en el despliegue.</li><li>2. Archivos Dockerfile correctamente configurados</li></ol>
Que el estudiante aprenda desplegar una arquitectura de software en la nube, utilizando servicios y herramientas de proveedores, con el fin de asegurar escalabilidad, disponibilidad y eficiencia en el uso de recursos, aplicando las mejores prácticas para la configuración y administración de entornos en la nube.	<ol style="list-style-type: none"><li>1. Proveedor de nube utilizado</li><li>2. Servicios de nubes aplicados</li><li>3. Componentes de la arquitectura configurados</li></ol>

El proyecto "Qnave" tiene como objetivo integrar y aplicar los conceptos adquiridos tanto en la clase magistral, como en el laboratorio del curso de Análisis y Diseño de Sistemas 2. En esta fase II, se requiere desarrollar una aplicación práctica basada en la documentación elaborada durante la fase I.

Para llevar a cabo esta fase, se deberá utilizar diversas herramientas de desarrollo que faciliten la implementación de los sistemas diseñados. Esto incluye la selección de Frameworks, estilos arquitectónicos, patrones de diseños y herramientas que le puedan ayudar para la gestión eficiente del proyecto.

### **¿Qué debe realizar en esta fase del proyecto?**

- 1. Rol Usuario**
  - 1.1. Registro
  - 1.2. Inicio de sesión
  - 1.3. Ver información del conductor
  - 1.4. Reportar algún problema
  - 1.5. Cancelar viaje
  - 1.6. Pedir viaje
- 2. Rol Conductor**
  - 2.1. Registro
  - 2.2. Iniciar sesión
  - 2.3. Aceptar viaje
  - 2.4. Cancelar viaje
  - 2.5. Reportar un problema
  - 2.6. Ver información del usuario
  - 2.7. Finalizar viaje (recibir pago)
- 3. Rol Asistente**
  - 3.1. Iniciar sesión
  - 3.2. Aceptar solicitudes de empleo (conductores)
  - 3.3. Ver información de conductores
  - 3.4. Dar de baja a un conductor
  - 3.5. Ver información de usuarios
  - 3.6. Dar de baja a usuario
- 4. Rol Administrador**
  - 4.1. Iniciar sesión
- 5. Actualizar documentación**
- 6. Documentación Sprint**
- 7. Tablero Kanban**

### **Descripción**

El CEO está plenamente satisfecho con la documentación que se le ha entregado, la cual refleja un trabajo meticuloso y bien estructurado por parte del equipo. Este documento no solo cumple con los estándares esperados, sino que también demuestra un entendimiento profundo de los desafíos y oportunidades que puede enfrentar la empresa.

Durante una reciente reunión, el CEO tuvo la oportunidad de profundizar en los detalles operativos de cada uno de los módulos que compondrán la aplicación, lo que ha permitido una mejor comprensión de cómo esta solución tecnológica impactará en las operaciones diarias de Qnave.

En esa discusión, se abordó de manera integral cómo cada módulo de la aplicación se integrará para optimizar tanto la eficiencia como la seguridad de nuestros servicios. El CEO expresó su tranquilidad al constatar que, más allá de los aspectos técnicos, se ha considerado cuidadosamente cómo cada componente

del sistema contribuirá a solucionar problemas críticos que hemos enfrentado, tales como la coordinación eficiente entre conductores y usuarios, así como la mejora en la seguridad, un aspecto fundamental para nuestra reputación y la confianza de nuestros clientes.

Es importante para el CEO que, si se diera el caso, en donde ocurriera algún problema al momento del desarrollo, **actualice** la documentación, ya que quiere tener todo por escrito de lo que se está realizando en el proyecto.

A continuación, se explica más a detalle cada una de las funciones que debe realizar en esta fase.

## 1. **ROL USUARIO**

### 1.1. **Registro del usuario**

Ya que anteriormente los conductores han sufrido robos y otros tipos de crímenes, es importante que la persona que desea el viaje esté totalmente identificando, por lo tanto, para poder usar la aplicación es necesario que el registro tenga la siguiente información:

- El nombre completo
- Fecha de nacimiento
- Género
- Correo
- Numero de celular
- Contraseña
- Confirmar contraseña

### 1.2. **Inicio de sesión**

El usuario podrá ingresar a su plataforma utilizando dos métodos:

- Utilizando el correo y la contraseña que se ingresaron al momento del registro.
- Utilizando el nombre de usuario que se le fue asignado al momento de haber sido registrado en la plataforma.

### 1.3. **Ver información del conductor**

Al momento que un conductor acepte el viaje, él usuario conseguirá visualizar la información del piloto donde podrá ver lo siguiente:

- Nombre del conductor
- Número de placa
- Foto del Automóvil
- Marca del Automóvil

### 1.4. **Reportar algún problema**

Es importante que el usuario esté seguro de cualquier percance que se dé, por lo tanto, es necesario contar con un módulo de “Reportar algún problema”, donde podrá exponer su situación.

- Si el problema fuera con algún conductor, él usuario debe ingresar la siguiente información:
  - a. Nombre
  - b. Número de placa
  - c. Descripción del problema
- Si el problema fuera de otra índole únicamente tiene que agregar una descripción y agregar la fecha desde cuando ha tenido este inconveniente.

### 1.5. **Cancelar viaje**

El usuario podrá cancelar el viaje, ya sea porque mucho tiempo de espera, ningún conductor ha aceptado el viaje o bien por cualquier otro motivo, si realiza esta acción debe explicar porque canceló el viaje, puede dar una descripción o bien puede mostrarle una serie de opciones.

### 1.6. **Pedir viaje**

El usuario podrá pedir viajes desde la aplicación indicando su punto de partida y punto de llegada, por cuestiones de tiempo, el administrador indica que los precios entre cada una de las zonas de la capital de Guatemala están fijos por el momento (ejemplo: de zona 1 a zona 12 tiene un precio

de viaje de Q 25.00) a continuación se muestra una tabla de cómo están cada una de las tarifas.

Punto de Partida A	Punto de Partida B	Tarifa
Zona 1	Zona 2	Q 10.00
Zona 1	Zona 3	Q 15.00
Zona 1	Zona 4	Q 20.00
Zona 1	Zona 5	Q 20.00
Zona 1	Zona 6	Q 25.00
Zona 1	Zona 7	Q 30.00
Zona 1	Zona 8	Q 30.00
Zona 1	Zona 9	Q 40.00
Zona 1	Zona 10	Q 50.00
Zona 1	Zona 11	Q 50.00
Zona 1	Zona 12	Q 25.00
Zona 2	Zona 3	Q 25.00
Zona 2	Zona 4	Q 25.00
Zona 2	Zona 5	Q 25.00
Zona 2	Zona 6	Q 30.00
Zona 2	Zona 7	Q 40.00
Zona 2	Zona 8	Q 40.00
Zona 2	Zona 9	Q 60.00
Zona 2	Zona 10	Q 70.00
Zona 2	Zona 11	Q 50.00
Zona 2	Zona 12	Q 35.00
Zona 3	Zona 4	Q 20.00
Zona 3	Zona 5	Q 30.00
Zona 3	Zona 6	Q 35.00
Zona 3	Zona 7	Q 25.00
Zona 3	Zona 8	Q 25.00
Zona 3	Zona 9	Q 40.00
Zona 3	Zona 10	Q 45.00
Zona 3	Zona 11	Q 40.00
Zona 3	Zona 12	Q 30.00
Zona 4	Zona 5	Q 15.00
Zona 4	Zona 6	Q 25.00
Zona 4	Zona 7	Q 25.00
Zona 4	Zona 8	Q 25.00
Zona 4	Zona 9	Q 15.00
Zona 4	Zona 10	Q 30.00
Zona 4	Zona 11	Q 35.00
Zona 4	Zona 12	Q 35.00
Zona 5	Zona 6	Q 15.00
Zona 5	Zona 7	Q 25.00
Zona 5	Zona 8	Q 35.00
Zona 5	Zona 9	Q 25.00
Zona 5	Zona 10	Q 35.00
Zona 5	Zona 11	Q 40.00
Zona 5	Zona 12	Q 40.00
Zona 6	Zona 7	Q 30.00
Zona 6	Zona 8	Q 35.00

<b>Zona 6</b>	Zona 9	Q 40.00
<b>Zona 6</b>	Zona 10	Q 50.00
<b>Zona 6</b>	Zona 11	Q 65.00
<b>Zona 6</b>	Zona 12	Q 60.00
<b>Zona 7</b>	Zona 8	Q 25.00
<b>Zona 7</b>	Zona 9	Q 35.00
<b>Zona 7</b>	Zona 10	Q 40.00
<b>Zona 7</b>	Zona 11	Q 40.00
<b>Zona 7</b>	Zona 12	Q 40.00
<b>Zona 8</b>	Zona 9	Q 25.00
<b>Zona 8</b>	Zona 10	Q 35.00
<b>Zona 8</b>	Zona 11	Q 35.00
<b>Zona 8</b>	Zona 12	Q 35.00
<b>Zona 9</b>	Zona 10	Q 15.00
<b>Zona 9</b>	Zona 11	Q 35.00
<b>Zona 9</b>	Zona 12	Q 30.00
<b>Zona 10</b>	Zona 11	Q 50.00
<b>Zona 10</b>	Zona 12	Q 50.00
<b>Zona 11</b>	Zona 12	Q 50.00

## 2. ROL CONDUCTOR

### 2.1. Registro

Los usuarios han reportado robos y otras ocasiones secuestros expés por parte de pilotos que “dicen” ser trabajadores de la empresa.

Por lo tanto, para que una persona pueda optar a un empleo (Piloto) en la compañía debe proporcionar la siguiente información.

- Nombre completo
- Número de teléfono
- Edad
- Número de DPI
- Correo electrónico
- Papelería completa (CV en formato PDF)
- Fotografía
- Fotografía del vehículo que usará para realizar viajes
- Número de placa
- Marca de Vehículo
- Año del vehículo
- Género
- Estado Civil
- Dirección de domicilio

**NOTA:** es importante tener un control sobre nuestros empleados, por esa misma razón a los conductores se les debe generar un código de empleado junto con una contraseña la cual se enviará a su correo electrónico para verificar que la cuenta con la que va a operar exista.

Ya que la contraseña que proporciona la empresa es temporal, es importante que al momento que la ingrese el asistente en la aplicación por primera vez este la cambie.

### 2.2. Iniciar sesión

Es conductor al igual que el usuario podrá ingresar a su plataforma utilizando dos métodos.

- Utilizando el correo y la contraseña (la contraseña es temporal, solamente servirá una vez, por lo tanto, debe ser la contraseña que el conductor ingreso al momento de entrar a la plataforma por primera vez).
- Puede ingresar utilizando su código de trabajador el cual se le asigna al momento de su registro en la plataforma.

### **2.3. Aceptar viaje**

Al momento que usuario mande la solicitud para un viaje este se enviara a todos los conductores. El sistema debe ser capaz de evitar inconveniente al momento de los viajes, eso quiere decir, que se tiene que ver en tiempo real si la solicitud del usuario ya fue respondida o bien, si aún ningún piloto ha aceptado el viaje, esto se busca evitar que dos conductores tengan en mismo pasajero al mismo tiempo.

### **2.4. Cancelar viaje**

El piloto puede cancelar un viaje solo si este fue aceptado con anterioridad, al igual que el usuario debe describir porque cancelo el viaje o bien se le dará una serie de opciones donde indicará la causa fue la razón de no prestar el servicio, si el conductor cancela más de diez viajes en un día se le bloqueará la cuenta y únicamente podrá activarla manda una justificación (Documento) donde explique la razón de porque ha cancelado tantos viajes. Esta opción se encontrará en la información general del conductor.

### **2.5. Reportar un problema**

El conductor podrá reportar cualquier inconveniente o problema que ocurra durante un viaje o relacionado con el uso de la plataforma. Para ello, contará con un formulario en el que deberá detallar la naturaleza del problema, seleccionar la categoría correspondiente (problemas con el pasajero, fallos técnicos, problemas de pago, etc.), y adjuntar cualquier evidencia relevante, como capturas de pantalla o fotos. Una vez enviado el reporte, el equipo de soporte de la plataforma evaluará la situación y proporcionará una respuesta en un plazo determinado. Este módulo es esencial para mantener la calidad del servicio y la seguridad tanto del conductor como del usuario.

### **2.6. Ver información del usuario**

Antes de aceptar un viaje, el conductor tendrá la opción de revisar la información básica del usuario que solicitó el viaje. Esta información incluye el nombre, la calificación general del usuario, el número de viajes completados, y cualquier comentario o alerta relevante dejada por otros conductores. Esta funcionalidad permite al conductor tomar decisiones informadas sobre la aceptación de un viaje, especialmente en situaciones donde puede haber dudas sobre la confiabilidad o seguridad del pasajero.

### **2.7. Finalizar viaje (recibir pago)**

Cuando el conductor llega al destino, debe indicar que el viaje ha terminado. La plataforma registrará la fecha y detalles del viaje. Además, es importante al momento de finalizado el viaje que él conductor indique si el pago fue hecho, sí dado caso no recibiera el pago, él conductor debe reportar al usuario.

## **3. ROL ASISTENTE**

Al inicio de las operaciones, debido a la limitada cantidad de vehículos disponibles, el administrador era el único responsable de gestionar las interacciones entre los usuarios y los conductores. Sin embargo, gracias a la gran popularidad que ha adquirido el servicio, reconocido por su efectividad en el ámbito del transporte privado, la demanda ha crecido considerablemente.

Como resultado de este éxito, el administrador ha asumido una serie de nuevas responsabilidades, lo que le impide continuar desempeñando ciertas funciones críticas, tales como la aceptación de solicitudes de nuevos conductores, la baja de conductores y usuarios, y el seguimiento de la información tanto de los usuarios como de los conductores para resolver posibles problemas surgidos durante los viajes.

Ante esta situación, el administrador ha decidido contratar asistentes que puedan brindar apoyo en la

gestión de la relación entre usuarios y conductores. Estos asistentes se encargarán de llevar a cabo las tareas que el administrador ya no puede realizar, asegurando que cualquier inconveniente que surja entre conductores y usuarios reciba el seguimiento adecuado y se mantenga la calidad del servicio ofrecido. A continuación, se describen más a detalle las funciones que tiene a su cargo:

### **3.1. Iniciar sesión**

- El asistente puede acceder a la plataforma usando su correo y contraseña (la contraseña es temporal, solamente servirá una vez, por lo tanto, debe ser la contraseña que el asistente ingreso al momento de entrar a la plataforma por primera vez), o con un código de trabajador asignado al registrarse.

### **3.2. Aceptar solicitudes de empleo (conductores)**

Debido a la inmensidad de usuarios y conductores, él administrador no puede operar la compañía solo, por lo tanto, él “**administrador**” solamente desde su plataforma podrá registrar este rol. La información que se debe ingresar es la siguiente:

- Nombre completo
- Número de teléfono
- Edad
- Numero de DPI
- Correo electrónico
- Papelería completa (CV)
- Fotografía
- Genero
- Estado Civil
- Número de cuenta
- Dirección de domicilio

**NOTA:** Al igual que los conductores, es importante tener un control sobre nuestros empleados, por esa misma razón también a los asistentes se les debe generar un código de empleado junto con una contraseña la cual se enviará a su correo electrónico para verificar que la cuenta con la que va a operar exista.

Ya que la contraseña que proporciona la empresa es temporal, es importante que al momento que la ingrese el asistente en la aplicación por primera vez este la cambie.

### **3.3. Ver información de los conductores**

El asistente podrá ver la información de los conductores, incluyendo su historial de viajes, calificaciones, comentarios de los usuarios, y el estado actual de su cuenta. Esto le permitirá tomar decisiones informadas al gestionar la relación con los conductores.

### **3.4. Dar de baja a un conductor**

El asistente podrá dar de baja a un conductor de la plataforma. Esto podrá hacerse en caso de incumplimiento de las políticas de la empresa, acumulación de quejas, o a petición del propio conductor. Al dar de baja a un conductor, se desactivará su cuenta y no podrá aceptar nuevos viajes. Se requerirá un motivo para realizar esta acción, que quedará registrado en el sistema.

### **3.5. Ver información de los usuarios**

Este módulo permitirá al asistente acceder a la información detallada de los usuarios registrados en la plataforma. Podrá ver datos como nombre, historial de viajes, comentarios recibidos, y cualquier otra información relevante para la administración del servicio.

### **3.6. Dar de baja a usuarios**

El asistente también podrá dar de baja a un usuario en caso de comportamiento inapropiado o incumplimiento de las políticas de uso de la plataforma. Al igual que con los conductores, se requerirá una justificación que se almacenará en el historial del usuario.



#### 4. ROL ADMINISTRADOR

##### 4.1. Iniciar sesión

Por temas de seguridad el inicio de sesión de administrador tiene que cumplir con lo siguiente:

- Se le otorgará un usuario y una contraseña la cual sólo él conocerá, esta información estará en la base datos encriptada.
- Después de ingresar las credenciales se nos enviará a otra pestaña, en donde la plataforma para verificar que si es él administrador el que está ingresando le pedirá un archivo de texto plano con extensión .ayd ejemplo (clave.ayd). Este archivo solo contendrá una segunda contraseña, es importante que esté encriptada. El sistema tiene que validar que lo ingresado en el documento de texto plano corresponda con la información de la base de datos, únicamente así podrá ingresar el administrador a su usuario.

##### **Observaciones importantes con registro e inicio de sesión.**

Cuando un usuario/conductor/asistente pierda su contraseña, tendrá la opción de recuperarla a través del proceso de "recuperación de contraseña". En este proceso, recibirá un mensaje en su correo electrónico, quedando a discreción del grupo la forma de cambio de contraseña.

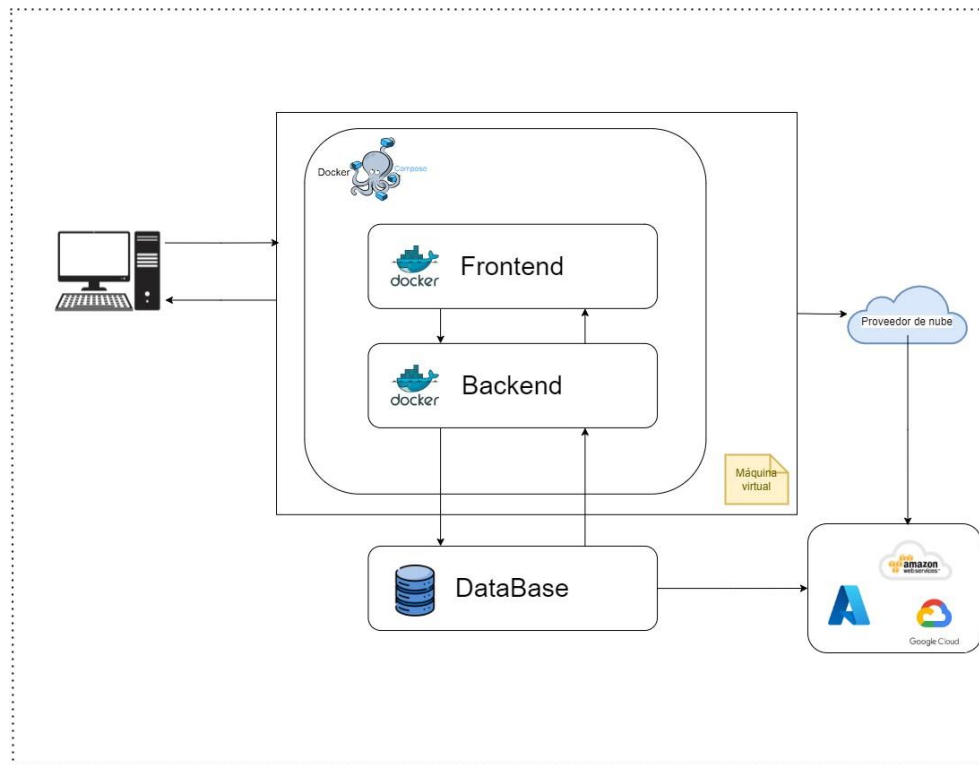
Cuando un usuario se registre, recibirá una notificación por correo electrónico solicitando la confirmación de la creación de su cuenta. Sin esta validación, el usuario no podrá utilizar la aplicación.

##### **Docker**

Permite crear, desplegar y gestionar aplicaciones en contenedores, los cuales son entornos de ejecución aislados que incluyen todo lo necesario para que una aplicación funcione: el código, las bibliotecas, las dependencias, y el sistema operativo base. A diferencia de las máquinas virtuales, los contenedores son más livianos y se inician mucho más rápido, ya que comparten el núcleo del sistema operativo subyacente.

Docker es fundamental en una arquitectura de software, ya que permite crear contenedores que garantizan la consistencia de las aplicaciones en diferentes entornos, eliminando problemas de compatibilidad y simplificando tanto el desarrollo como el despliegue. Al encapsular todas las dependencias necesarias, Docker asegura que las aplicaciones funcionen de manera idéntica en desarrollo, pruebas y producción, y proporciona portabilidad al permitir que los contenedores se ejecuten en cualquier sistema con Docker. Además, facilita la escalabilidad y la gestión eficiente de recursos al permitir la ejecución de múltiples contenedores en un solo servidor, optimizando así el uso de recursos y mejorando la capacidad de respuesta ante la demanda.

Para esta fase se le solicita al estudiante que utilice Docker tanto en el Backend como en el Frontend. A continuación, se le presenta un ejemplo de como seria la forma de utilizar Docker.



El proveedor de la nube queda a discreción del grupo cuál usar (AWS, Azure o GCP).

### Tablero Kanban

El tablero Kanban se utiliza para llevar un control de las tareas que deben de realizarse a lo largo del proyecto para que todo el grupo de desarrolladores visualice el progreso de este.

Se debe hacer uso de herramientas de gestión de proyectos para llevar el control de las tareas por medio de la creación de un tablero Kanban en donde se debe de tener como mínimo las siguientes columnas:

- To Do
- Blocked
- in progress
- Ready for Testing
- Test/QA
- Done

### Estrategia de Branching

Para esta fase se debe realizar la estrategia de branching git-flow, están las ramas principales llamadas main y develop, por otra parte, están las ramas de soporte que son feature, reléase y hotfix.

Cada estudiante debe tener su rama feature llamada feature#Carnet. Se debe hacer un merge a la rama main con el nombre de la versión del reléase que en este caso sería (v1.0.0) con su respectivo tag.

### Documentación Sprint

Se debe de implementar durante el desarrollo del proyecto como mínimo 2 sprints. Tras la finalización de cada Sprint deben de redactar un documento en un archivo con formato markdown que conste de los siguientes elementos:

- Daily Scrum en donde CADA INTEGRANTE indica que trabajará en cada daily, que hizo el día anterior y si tiene algún impedimento.
- Tabla en donde se observen los elementos del Sprint Backlog que se logró terminar y los que no con justificación para los que no se lograron terminar.

- imagen en donde se visualiza el tablero previo al inicio del sprint.
- imagen en donde se visualiza el tablero en la finalización del sprint.
- Sprint planning donde deben de indicar que es lo que se trabajará en el sprint
- Sprint retrospective: CADA UNO DE LOS INTEGRANTES debe de responder las preguntas:
  - ¿Qué se hizo bien durante el Sprint?
  - ¿Qué se hizo mal durante el Sprint?
  - ¿Qué mejoras se deben implementar para el próximo sprint?

**NOTA:**

Es importante tomar en cuenta que cada sprint es de 15 días por lo tanto como mínimo debe contar con 6 daily en su documentación.

Tanto para el sprint planning como para el retrospective se debe realizar una grabación donde todos los participantes estén colaborando.

**CONSIDERACIONES:**

1. Los cambios realizados en esta fase se tienen que ver reflejado en la documentación.
2. Todo lo realizado en su documentación tiene que estar implementado en esta fase.
3. Es obligatorio el uso de una estrategia de **branching**.
4. El framework, así como el lenguaje de programación tanto para el desarrollo de Backend como el de Frontend queda a discreción del grupo.
5. Todo debe estar en la nube, **no se calificará nada local**.
6. Aplicar buenas prácticas para el uso de los commits.
7. **Las copias parciales o totales tendrán una nota de 0 y será reportada a la escuela de sistemas.**
8. Todos los integrantes deben estar presentes el día de la calificación, ya que se validará la participación de los integrantes.
9. Es necesario emplear una base de datos para almacenar la información, la cual debe estar alojada en la nube. Esta base de datos puede ser relacional o no relacional, según las necesidades del sistema y la naturaleza de los datos a gestionar.
10. Agregar al auxiliar al repositorio del grupo: “Mau-Noj”
11. El auxiliar todo el tiempo debe de tener acceso al repositorio. Si no se tiene acceso al repositorio al momento de la calificación se tendrá una penalización.
12. Para los datos sensibles como contraseñas se le solicita al estudiante que cifre los datos, la herramienta queda a discreción del grupo.
13. La documentación debe ser almacenada en el repositorio utilizando el formato markdown.
14. Almacenamiento de imágenes, deberán utilizar Google cloud storage o un bucket de Amazon S3.
15. La visualización de los archivos PDF es flexible y se puede realizar utilizando cualquier herramienta disponible.

## Puntos que abarcaran esta fase del proyecto en relación con el contenido del curso (laboratorio y clase magistral)

### Laboratorio

#### 1. Frameworks

- 1.1. Qué es un framework
- 1.2. Tipos de framework
- 1.3. Beneficios del uso de frameworks
- 1.4. Características comunes de los frameworks
- 1.5. Casos de uso específicos
- 1.6. Diferencias entre framework y Librería

#### 2. Elementos de diseño

- 2.1. Patrones de diseño
  - 2.1.1. Definición de patrones de diseño
  - 2.1.2. Importancia y aplicabilidad de patrones de diseño
  - 2.1.3. Notaciones de patrones de diseño
  - 2.1.4. Ejemplos de patrones de diseño
- 2.2. Contexto del diseño
- 2.3. Principios de diseño
- 2.4. Tipos de patrones de diseño
  - 2.4.1. Creacional
    - 2.4.1.1. Singleton
    - 2.4.1.2. Abstract Factory
    - 2.4.1.3. Factory Method
  - 2.4.2. Estructural
    - 2.4.2.1. Adapter
    - 2.4.2.2. Decorator
    - 2.4.2.3. Facade
    - 2.4.2.4. Proxy
  - 2.4.3. Comportamiento
    - 2.4.3.1. Observer
    - 2.4.3.2. Strategy
    - 2.4.3.3. Command
    - 2.4.3.4. Iterator
- 2.5. Catálogo de estilos
- 2.6. Experiencia de usuario (UX)
- 2.7. Diseño de interfaces de usuario (UI)

### Magistral

#### 5. Contenido temático del curso

Unidad	Tema
1. Principios de Diseño de Arquitectura de Software	1. ¿Qué es Arquitectura de Software?
	1.1. Concepto
	1.2. ¿Por qué es importante?
	1.3. Beneficios de una Arquitectura de SW
	1.4. Pasos para la definición de una Arquitectura
	1.5. ¿Qué se tiene en cuenta para el diseño?
	1.6. ¿Cómo se modela?
	1.7. Estructuras y Vistas
	1.4.1 Categorías de Estructuras
	1.8. Generos y Estilos Arquitectónicos
	1.8.1 Arquitectura Candidata
	1.8.2 La Arquitectura de Referencia
	1.8.3 Diseño Arquitectónico
	1.8.4 Catálogo de Estilos
	1.8.4.1 Cliente-Servidor
	1.8.4.2 Centrada en Datos
	1.8.4.3 En o por capas
	1.8.4.4 Centrada en el flujo de datos
	1.8.4.5 Llamada y retorno
	1.8.4.6 Basada en eventos
	1.8.4.7 Modelo-Vista-Controlador