

# PROYECTO ÚNICO

## MONEY BIN BANK

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: "P"



Guatemala 09 de diciembre  
Vacaciones de Segundo Semestre 2024

# OBJETIVOS GENERALES

- Aplicar los conocimientos adquiridos tanto en la clase magistral como en el laboratorio.
- Diseñar una arquitectura de software adaptable para aplicaciones empresariales, aplicando diversos principios de diseño de software para comprender la importancia de una estructura flexible y escalable que facilite la evolución y el mantenimiento de la arquitectura mediante buenas prácticas.
- 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.
- Comprender la importancia y la organización de trabajo en equipo.

# OBJETIVOS ESPECIFICOS

- Analizar un problema de software, identificando los patrones de diseño más adecuados para su solución y desarrollando una implementación que los utilice de manera efectiva.
- Aplicar y resolver problemas de diseño relacionados con la distribución de software en hardware, utilizando diagramas de despliegue para visualizar y comunicar soluciones efectivas.
- Comprender cómo el análisis de los requisitos del sistema permite comparar diferentes estilos arquitectónicos y seleccionar el más adecuado, justificando la elección en función de criterios técnicos y de negocio.
- Modelar el comportamiento del sistema mediante casos de uso bien definidos, asegurando que se cubran todos los actores, precondiciones, postcondiciones, flujos alternativos y excepciones.
- Identificar de forma precisa los requisitos funcionales y no funcionales de soluciones de software, logrando una comprensión clara de las necesidades del usuario.
- 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.

- 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.
- Entender la importancia de los diferentes tipos de pruebas en una arquitectura de software con el objetivo de identificar errores, validar el funcionamiento correcto de los diferentes módulos y garantizar la calidad general del producto final, asegurando un desarrollo efectivo y exitoso.
- Comprender la importancia del uso de Terraform y Ansible en la implementación de Infraestructura como Código (IaC) dentro de una arquitectura de software, para gestionar y aprovisionar de manera automatizada y declarativa la infraestructura de TI, optimizando la configuración, el despliegue y el mantenimiento con mayor eficiencia, consistencia y control, contribuyendo así a la estabilidad y escalabilidad del sistema.
- Comprender la importancia de CI/CD en la arquitectura de software con el objetivo de automatizar y estandarizar la integración y entrega de aplicaciones, permitiendo implementar cambios de código de manera fluida y segura en entornos de producción, mejorando la resiliencia y escalabilidad de la arquitectura mientras se minimizan riesgos y se facilita una respuesta ágil ante cambios o incidentes.

# DESCRIPCIÓN

El CEO ha expresado su satisfacción con el trabajo realizado hasta ahora, lo que ha llevado a la decisión de encargarle a usted y a su equipo el desarrollo de una plataforma para el Banco "Money Bin". Es fundamental que tenga en cuenta que Money Bin se posiciona como una institución financiera moderna, ágil y accesible, cuyo enfoque está en ofrecer soluciones financieras seguras y personalizadas tanto para individuos como para empresas. Su misión es promover la inclusión financiera y contribuir al crecimiento económico de sus clientes a través de servicios innovadores. La creación de esta plataforma no solo debe alinearse con los valores del banco, sino que también debe reflejar un compromiso genuino con la innovación y la adaptabilidad. En un entorno financiero en constante evolución, es crucial que la solución que desarrolle no solo cumpla con las expectativas actuales, sino que también anticipe las necesidades futuras de los usuarios. Esto implica integrar tecnologías avanzadas y asegurar una experiencia de usuario fluida y eficiente. El éxito de Money Bin dependerá en gran medida del enfoque que usted y su equipo adopten en este proyecto. La capacidad de la plataforma para facilitar el acceso a servicios financieros, fomentar la confianza del cliente y proporcionar una experiencia personalizada será determinante para atraer y retener a los usuarios. Además, es esencial considerar cómo la plataforma puede contribuir a la inclusión financiera, permitiendo que más personas y empresas accedan a productos y servicios financieros que antes les eran inaccesibles.

El CEO del banco le ha proporcionado la siguiente información de cada uno de los distintos roles que existen:

## ROL CAJERO

### LOGIN

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.

### PAGO DE SERVICIOS

El cajero puede realizar pagos de servicios básicos en nombre de los clientes desde el sistema del banco. Este módulo facilita la gestión de pagos de servicios tales como:

- Agua
- Luz
- Teléfono
- Internet

Los datos que pedirá el sistemas para poder realizar los pagos son los siguientes:

- Nombre de la persona encargada del servicio
- Código de servicio
- monto que debe pagar

Otra manera que puede el cliente pagar es desde su cuenta bancaria, en este caso los datos a ingresar serían:

- Nombre de la persona encargada del servicio
- Código de servicio
- monto que debe pagar
- Número de cuenta
- Número de DPI de la persona dueña de la cuenta

Es importante validar que la persona que esta haciendo el pago sea la persona titular de la cuenta.

## **PAGO DE PRESTAMOS**

Permite a los clientes realizar pagos parciales o totales de préstamos adquiridos con el banco Money Bin. Este módulo asegura que los pagos sean registrados adecuadamente y aplicados a los saldos de préstamo correspondientes. Debe contar con los siguientes campos:

- Número de cuenta del cliente
- Número de préstamo
- Monto a pagar
- Fecha de pago

## **BUSCAR POR CUENTA O POR CUI**

Dado el gran número de clientes, es crucial poder localizar rápidamente la cuenta de un cliente mediante su número de cuenta o CUI. El módulo permite a los empleados:

- Buscar a un cliente usando su número de cuenta o CUI
- Visualizar el perfil del cliente con detalles de contacto e historial de transacciones

## **MOSTRAR SALDO**

Este módulo permite a los empleados y al cliente consultar el saldo disponible en sus cuentas. Los detalles mostrados incluyen:

- Saldo actual
- Fecha de la última actualización de saldo

## **CAMBIAR MONEDA (QUETZALES A DÓLARES)**

El cliente puede hacer cambio de quetzales a dólares, es importante validar el limite de cambio que puede hacer al día el cual será de Q 10,000.00 y esto solo lo puede hacer una vez al mes.

## **GENERAR COMPROBANTE**

Para cada transacción realizada, el sistema debe generar un comprobante en formato PDF, proporcionando a los clientes un registro de sus operaciones bancarias. El comprobante debe incluir:

1. Número de cuenta (Solo si el cliente paga con su cuenta)
2. Tipo de transacción (pago de servicios, préstamo, depósito, etc.).
3. Fecha y hora
4. Monto
5. Nombre y firma digital del empleado autorizado

## **RETIRO DE DINERO**

Permite a los clientes realizar retiros de efectivo desde sus cuentas. El módulo incluye funciones de validación y registro, y debe contar con la siguiente información:

1. Número de cuenta del cliente
2. Monto a retirar
3. Fecha y hora del retiro
4. Si es monetario o ahorro
5. Si es en Quetzales o Dólares
6. tipo de cuenta

Nota: Se deben implementar límites de retiro y validaciones de identidad para evitar fraudes.

## **DEPOSITAR DINERO**

Este módulo permite a los clientes realizar depósitos en sus cuentas. Los depósitos se registran en tiempo real y actualizan el saldo disponible. Se requiere la siguiente información:

1. Número de cuenta
2. Monto a depositar
3. Fecha y hora del depósito
4. Si es monetaria o de ahorro
5. Si el deposito será en dólares o quetzales.
6. tipo de cuenta

Para hacer depósitos en dólares es importante que la persona ya cuente con este modulo activo de lo contrario no lo dejara depositar.

## **PAGO DE TARJETA DE CRÉDITO**

El cliente podrá pagar su tarjeta de crédito en el cajero, debe tomar en cuenta que al mes se le cobra Q 75.00 por solo tener la tarjeta, y a eso debe agregarle el dinero que a cargado a su tarjeta. si la persona no paga a tiempo se le cobrara un interés simple del 5% de su deuda.

# **ROL ATENCIÓN AL CLIENTE**

## **LOGIN**

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.

## **CREACIÓN DE CUENTA (MONETARIO O AHORRO)**

Permite a los clientes abrir nuevas cuentas de ahorro o monetarias. Este módulo incluye validaciones y debe recoger:

- Información personal del cliente.
  - Nombre
  - Apellido
  - CUI
  - Teléfono
  - Correo electrónico
  - Edad
  - Género
  - Fotografía
- Tipo de cuenta (Monetario o Ahorro).
- Agregar una pregunta de seguridad con su respectiva respuesta (las cuales debe estar encriptada)
- Monto con el que abrirá su cuenta
- Fecha y hora de apertura.

Nota: Debe realizarse una validación de identidad y generación automática de un número de cuenta único.

## **ACTUALIZACIÓN DE DATOS DEL CLIENTE**

Permite a los clientes modificar su información personal para mantenerla actualizada. Este módulo debe incluir validaciones y permisos para garantizar la seguridad de los datos, y debe incluir:

- Número de cuenta.
- Datos a actualizar (Teléfono, Dirección, Correo Electrónico, etc.).
- Fecha y hora de la actualización.
- Verificación de identidad (pregunta de seguridad).

Nota: Se debe guardar en un historial todos los cambios que realicen los clientes.

## **CREAR NUEVAS TARJETAS (CRÉDITO O DÉBITO)**

Permite a los clientes solicitar nuevas tarjetas de crédito o débito. Este módulo debe garantizar la validación de identidad y el cumplimiento de políticas del banco, incluyendo:

- Número de cuenta único
- Tipo de tarjeta (Crédito o Débito).
- Límite de crédito solicitado (si aplica).
- Nombre del titular
- Fecha y hora de la solicitud.
- Estado de la solicitud (Pendiente, Aprobada, Rechazada).

Nota: Para que el cliente pueda optar a una tarjeta esta debe ser aprobada por el supervisor.

## **BLOQUEO DE TARJETAS**

Permite a los clientes bloquear sus tarjetas en caso de robo, pérdida o sospecha de fraude. Este módulo debe incluir:

- Número de tarjeta o a del titular (validando con la pregunta de seguridad)
- Motivo del bloqueo (Robo, Pérdida, Fraude).
- Fecha y hora del bloqueo.
- Tipo de tarjeta (Crédito o Débito).

Nota: El módulo debe notificar al cliente vía correo electrónico sobre el bloqueo exitoso y generar un reporte automático al equipo de fraude.

## **SOLICITUDES DE CANCELACIÓN DE SERVICIOS**

Permite a los clientes solicitar la cancelación de servicios o productos financieros. Este módulo debe incluir:

- Número de cuenta o identificación.
- Servicio a cancelar (Cuenta o Tarjeta).
- Motivo de la cancelación.
- Fecha y hora de la solicitud.
- Estado de la solicitud (Pendiente, Procesada, Rechazada).

Nota: El banco debe confirmar la identidad del cliente y revisar saldos o deudas pendientes antes de la cancelación, por lo tanto la cancelación únicamente la podrá hacer el supervisor después de haber examinado el perfil del cliente.

## **PERMITIR CUENTAS EN DÓLARES**

Permite a los clientes abrir cuentas en dólares como alternativa a cuentas en quetzales. Este módulo debe incluir:

- Número de cuenta.
- Tipo de cuenta (Monetario o Ahorro).
- Fecha y hora de apertura.

Nota: Este modulo lo podrá usar el cliente para depósitos o retiros solo si solo haya realizado este tramite primero.



## **ENCUESTAS DE SATISFACCIÓN**

Permite a los clientes evaluar los servicios del banco a través de encuestas. Este módulo debe incluir:

- Número de cliente o identificación (opcional).
- Categoría evaluada (Atención al Cliente, Servicios, Productos, etc.).
- Fecha y hora de la encuesta.
- Puntuación y comentarios del cliente.

Nota: Los datos deben ser almacenados para análisis estadístico y retroalimentación. Esta encuesta la hará el personal de atención al cliente, a todos los clientes.

## **SOLICITUD DE PRÉSTAMOS AL BANCO**

Permite a los clientes solicitar préstamos personales, hipotecarios, vehiculares u otros. Este módulo debe garantizar un proceso transparente y seguro, incluyendo:

- Número de cuenta o identificación.
- Tipo de préstamo solicitado: Personal, Hipotecario, Vehicular, Educativo, etc.
- Monto solicitado.
- Plazo del préstamo: En meses o años.
- Fecha y hora de la solicitud.
- Estado de la solicitud: Pendiente, Aprobado, Rechazado.
- Documentación requerida: Comprobantes de ingresos, garantías, identificación, etc. (todo en formato PDF).

Nota:

- 1.El sistema debe realizar un análisis automático de la capacidad de pago del cliente, verificando si no tiene alguna deuda actual.
- 2.Deben establecerse tasas de interés dinámicas basadas en el historial crediticio del cliente y el tipo de préstamo.
- 3.Las solicitudes rechazadas deben incluir una justificación clara y recomendaciones para el cliente.
- 4.El Encargado de dar autorización de los prestamos es el supervisor.

## **REGISTRO DE QUEJAS**

Permite a los clientes registrar sus quejas sobre los servicios del banco. Este módulo incluye funciones para categorizar y priorizar las quejas y debe incluir la siguiente información:

- Número de cuenta o identificación.
- Detalles de la queja.
- Fecha y hora del registro.
- Tipo de queja (Servicio, Producto, Atención al Cliente, etc.).

Nota: Debe enviar un correo al supervisor sobre toda esta información.

# **ROL ADMINISTRADOR DE SISTEMAS**

## **REGISTRO EMPLEADO**

El administrador de sistemas será el encargado de registrar nuevos trabajadores al sistemas, para su registro se le pedirá la siguiente información:

- Nombre completo
- Número de teléfono
- Edad
- Numero de DPI
- Correo electrónico
- Papelería completa (PDF)
- Fotografía
- Genero
- Estado Civil

Al momento de su registro se le enviara a su correo una confirmación de sus datos, además, también se le mandara un nombre de usuario UNICO junto con una contraseña la cual genera de forma automática teniendo 16 caracteres llevando símbolos, letras, números, todo con tal sea lo más seguro posible, además todo esto debe estar encriptado en la base de datos.

## **ASIGNACIÓN DE ROLES**

El administrador de sistemas será el encargado de asignarlos los roles a los empleados después de su registro, por lo tanto el rol que se le asigne a un nuevo empleado puede cambiar por temas de rotación.

## **ELIMINAR EMPLEADOS**

Él administrador de sistemas debe poder eliminar a los empleados únicamente cuando el supervisor lo indique, por lo tanto, desde la plataforma del supervisor se debe mandar un archivo PDF en donde explicara a detalle la razón por la cual se esta dando de baja al empleado y este documento se puede visualizar desde la plataforma, luego de visualizar toda la información el administrador de sistemas procedera

## **CAMBIO DE CONTRASEÑA EMPLEADO**

Él administrador de sistemas le puede proporcionar una nueva contraseña al empleado, solo podrá hacer el cambio cuando el supervisor se lo indique, de lo contrario no puede cambiar ninguna contraseña.

## **GESTIÓN DE COPIAS DE SEGURIDAD**

Él administrador de sistemas puede generar una copia de seguridad de toda la base de datos del banco, es de suma importancia que esta copia quede resguarda en un servicio en la nube, y que cualquier anomalía o eventualidad que se pueda dar este disponible las 24 horas del día.

# **SUPERVISOR**

## **LOGIN**

Por temas de seguridad el inicio de sesión de Supervisor 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 le enviará a otra pestaña, en donde la plataforma para verificar que si es él administrador es la persona que está ingresando, donde se 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

## **MODIFICAR INFORMACIÓN DE EMPLEADO**

El empleado puede solicitar al supervisor el cambio de algún aspecto de su información, donde el supervisor le solicitara lo siguiente:

- Nombre único de usuario o CUI

Después el programa al supervisor toda la información del empleado donde únicamente los datos que puede cambiar son los siguientes:

- Nombre (No se puede cambiar apellido)
- Número de teléfono
- Correo electrónico
- Fotografía
- Genero
- Estado Civil

Debe realizar las validaciones pertinentes para evitar que cualquier dato que no aparezca no sea modificado. Si en dado caso el empleado lo que necesita es un cambio de contraseña, el supervisor desde su plataforma le mandara una notificación al administrador de sistemas para que le proporcione otra; en el caso que el empleado haya perdido mas de tres veces la contraseña este automáticamente será despedido, por lo tanto ya no tendrá acceso al sistema.

## **ACEPTAR PRESTAMO**

El supervisor podrá aceptar los prestamos que los clientes soliciten, toda la información referente al préstamo la podrá visualizar en su pantalla, lo único que hará el sistema de forma automática es validar que la persona no tenga un préstamo pendiente, si se diera el caso que si tenga un préstamo activo aún se petición será rechaza de forma automática.

## **ENCUESTAS DE SATISFACCIÓN**

El supervisor podrá ver toda la respuestas que han proporcionado los clientes al momento de evaluar los servicios. Esto lo hace con el fin de buscar puntos de mejora y hacer que la empresa pueda tener un acercamiento a sus clientes. Las preguntas que se deben hacer en la encuesta quedan a discreción del grupo cuales realizar, debe ser un mínimo de 5 y un máximo de 10.

## **VER REGISTRO DE QUEJAS**

El supervisor podrá ver todas las quejas que las personas han realizado sobre algún servicio, esto con el fin de evaluar futuros puntos de mejora.

## **REGISTRAR ADMINISTRADOR DE SISTEMAS**

El supervisor será el encargado de registrar a u nuevo administrador de sistemas, para su registro se le pedirá la siguiente información:

- Nombre completo
- Número de teléfono
- Edad
- Numero de DPI
- Correo electrónico
- Papelería completa (PDF)
- Fotografía
- Genero
- Estado Civil

Al momento de su registro se le enviara a su correo una confirmación de sus datos, además, también se le mandara un nombre de usuario UNICO junto con una contraseña la cual genera de forma automática teniendo 16 caracteres llevando símbolos, letras, números, todo con tal sea lo más seguro posible, además todo esto debe estar encriptado en la base de datos. Ya que el administrador de sistemas cuenta con aspectos importantes para la empresa, es necesario que cuente con autenticación de dos factores (2FA), el cual queda a discreción del grupo como realizarlo.

## **INFORMACIÓN DE LOS ADMINISTRADOR DE SISTEMAS**

El supervisor podrá modificar, eliminar o ver la información de cada uno de los administradores de sistemas, es importante que esta información resguarde la integridad del empleado, por lo tanto cualquier cambio que se realice este será notificado vía correo electrónico sobre las acciones que han realizado en su perfil.

## **MONITOREO DE ACTIVIDAD EN TIEMPO REAL**

Permite al supervisor observar en tiempo real las actividades realizadas en el sistema por empleados y administrador de sistemas. Este módulo debe incluir:

- Seguimiento en tiempo real de:
  - Retiros de dinero.
  - Solicitudes de préstamos.
  - Bloqueo/Activación de tarjetas.
  - Ver algunas las actividades que realiza el administrador de sistemas
    - .registro empleado
    - asignación de roles
    - eliminar empleados
    - cambio de contraseña empleado
- Filtros de búsqueda por: Empleado, Fecha, Tipo de actividad.
- Alertas automáticas: Para actividades sospechosas o incumplimientos de políticas, esta notificación se enviara al correo del supervisor.

Nota: Se debe almacenar un registro detallado para auditorías posteriores.

## **GESTIÓN DE INVENTARIOS BANCARIOS**

Este módulo permite a los supervisores controlar el efectivo y otros recursos disponibles en la sucursal para garantizar una administración eficiente y segura. Se lleva un registro detallado de entradas y salidas diarias, con funciones de monitoreo en tiempo real y generación de reportes automáticos.

Deberá mostrar la siguiente información.

- Dinero que dispone el Banco (Tanto en dólares como en quetzale)
- Ver todas las entradas y salidas de dinero (Debe agregar filtros)
- Ver las ganancias o perdidas a lo largo del tiempo.

## **APROBAR TARJETAS (CRÉDITO O DÉBITO)**

El supervisor será el encargado de aprobar la solicitudes de los clientes para optar a una tarjeta, la forma que el tomara la decisión será mostrando toda la información del cliente, así como su récord financiero en la institución.

## **APROBAR CANCELACIÓN DE SERVICIOS**

El supervisor será responsable de aprobar las solicitudes de los clientes para cancelar servicios. Sin embargo, antes de proceder con la baja, deberá verificar que no existan inconvenientes, como saldos pendientes en la cuenta. Para prevenir posibles casos de corrupción, es fundamental que el sistema impida al supervisor cancelar una cuenta si detecta actividad sospechosa.

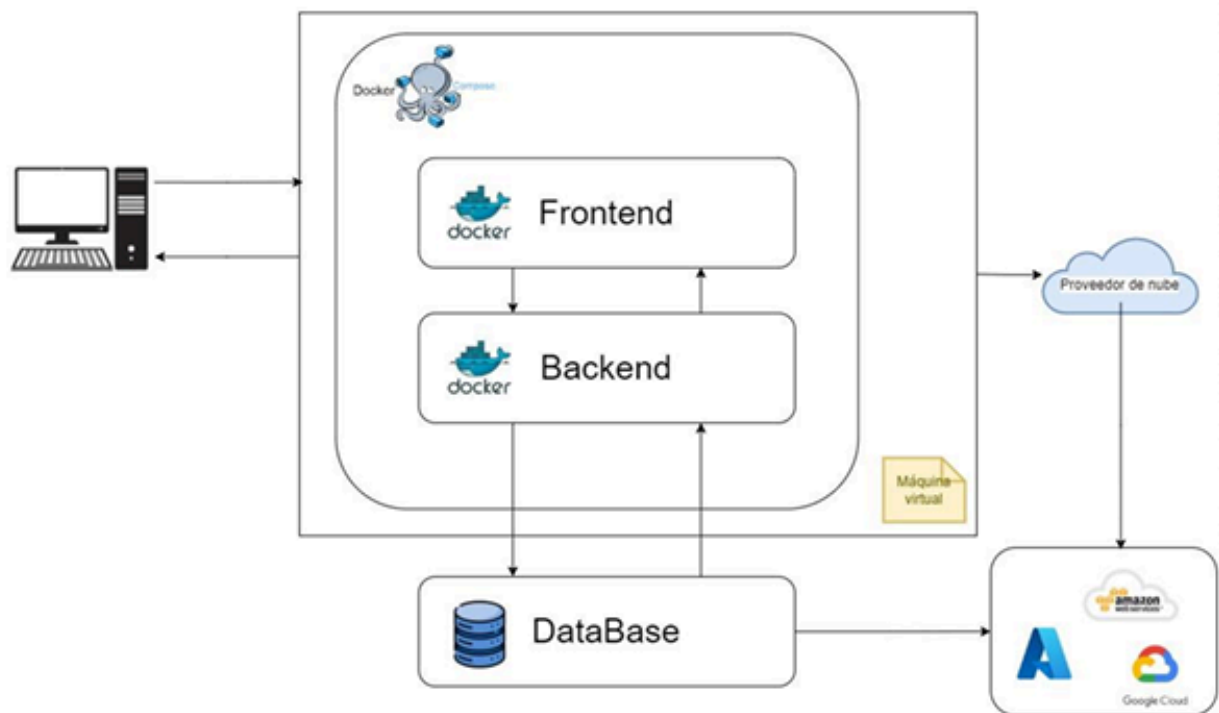
## **REPORTES**

El supervisor tendrá acceso a reportes , los cuales le ayudaran a tomar distintas decisiones en el banco. Los reportes quedan a discreción del grupo cuales hacer, pero siempre tomando en consideración que deben darle valor al negocio. debe hacer un mínimo de tres reportes.

# DOCKER

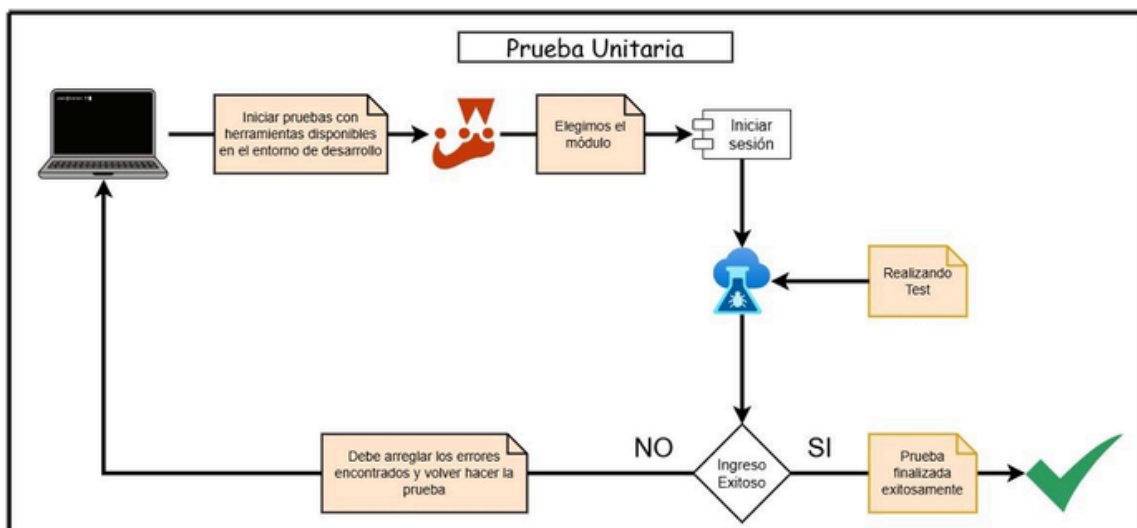
Permite crear, desplegar y gestionar aplicaciones en contenedores, los cuales son entornos de ejecución aislados que incluyen todo lo necesario para que una aplicación funcione: el código, las bibliotecas, las dependencias, y el sistema operativo base. A diferencia de las máquinas virtuales, los contenedores 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.



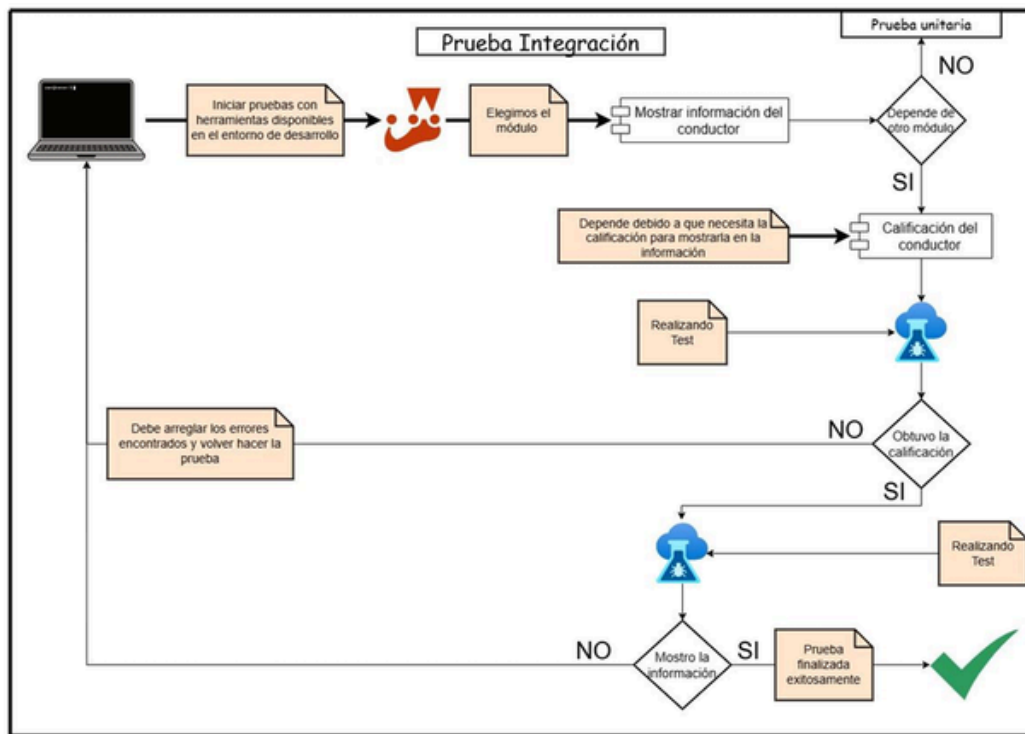
# PRUEBAS UNITARIAS

Las pruebas unitarias se pueden definir simplemente como un procedimiento que se ejecuta para validar que cada unidad o componente individual de una aplicación funcione correctamente. Este tipo de pruebas confirma que cada módulo en aislamiento produce el resultado esperado dado un conjunto de entradas. Por ende, se verifican partes específicas del código sin interacción con dependencias externas. Tiene opciones a utilizar como Jest, Mocha o cualquier Framework que se adapte a su proyecto. Para realizar pruebas unitarias y asegurar que cada función individual trabaje según lo esperado. Deberá implementar un mínimo de 5 pruebas para cubrir todos los escenarios posibles



# PRUEBAS DE INTEGRACIÓN

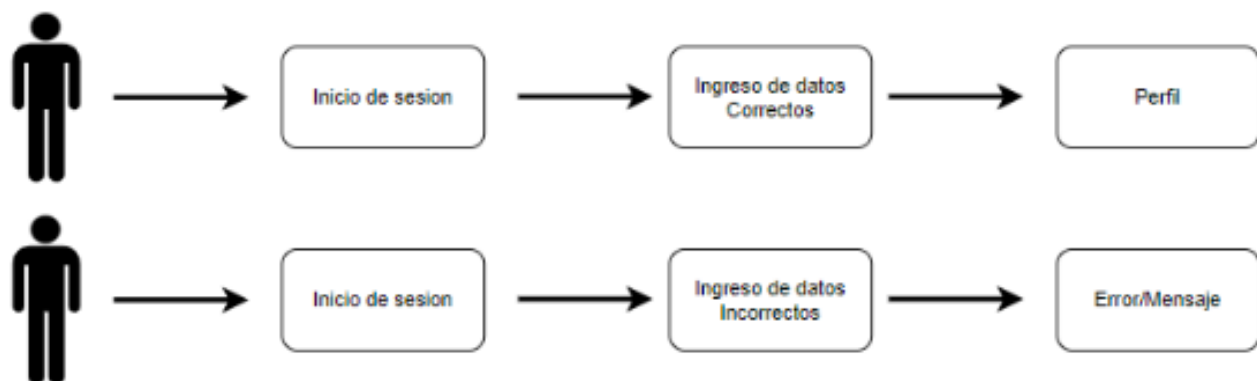
Las pruebas de integración se pueden definir simplemente como un procedimiento que se ejecuta para verificar la interacción entre diferentes módulos o componentes de una aplicación. Este tipo de pruebas confirma que los módulos integrados funcionan juntos como se espera, detectando posibles errores en las interfaces y flujos de datos entre ellos. Por ende, se valida la correcta comunicación y coordinación entre los componentes que se combinan para realizar funciones más complejas. Tiene opciones de herramientas como Jest, Mocha, pytest o cualquier otra herramienta para realizar pruebas de integración, asegurando que las distintas partes del sistema se integren de manera fluida y correcta. Deberá implementar un mínimo de 5 pruebas.



## PRUEBAS DE ACEPTACIÓN

Las pruebas de aceptación se pueden definir simplemente como un procedimiento que se ejecuta para validar que una aplicación cumple con los requisitos especificados y satisface las necesidades del usuario final. Este tipo de pruebas confirma que el sistema completo funciona como se espera en un entorno de producción, verificando la funcionalidad desde la perspectiva del usuario. Por ende, se evalúa la conformidad del software con los criterios de aceptación definidos previamente. Deberá implementar un mínimo de 5 pruebas que cubran los casos de uso más importantes del usuario.

## PRUEBAS E2E



Las pruebas end-to-end se pueden definir simplemente como un procedimiento que se ejecuta para productos complejos. Este tipo de pruebas confirman que la aplicación funciona tal como se espera analizando todos sus componentes. Por ende, se simula la experiencia del usuario de principio a fin. Deberán utilizar selenium o cypress para realizar pruebas end-to-end para simular la interacción del usuario con el UI. Deberá implementar un mínimo de 5 pruebas.



# **PRUEBAS NO FUNCIONALES**

## **PRUEBAS DE SEGURIDAD (PENTESTING)**

Las pruebas de seguridad o pentesting se pueden definir como un procedimiento ejecutado para evaluar y asegurar la protección de una aplicación frente a vulnerabilidades o ataques externos. Estas pruebas simulan técnicas de ataque reales para identificar debilidades en los sistemas, verificando la resistencia a amenazas como accesos no autorizados, fugas de información o violaciones de datos. El objetivo es garantizar que la aplicación cumpla con altos estándares de seguridad. Para realizar estas pruebas, se pueden utilizar herramientas como Metasploit, Burp Suite, OWASP ZAP y Nmap.

## **PRUEBAS DE RENDIMIENTO**

Las pruebas de rendimiento consisten en evaluar cómo se comporta una aplicación bajo diversas cargas de trabajo para garantizar su eficiencia y estabilidad. Este tipo de pruebas mide factores críticos como el tiempo de respuesta, la capacidad de procesamiento y el uso de recursos cuando se ejecutan operaciones simultáneas. Detecta cuellos de botella y asegura que el sistema pueda manejar el tráfico esperado. Herramientas como JMeter, Gatling, LoadRunner y k6 son útiles para ejecutar pruebas de carga, estrés y escalabilidad.

## **PRUEBAS DE INYECCIÓN SQL**

Las pruebas de inyección SQL son procedimientos diseñados para detectar vulnerabilidades relacionadas con la manipulación de consultas SQL en una aplicación. Estas pruebas buscan identificar puntos donde un atacante podría insertar código malicioso para acceder a datos confidenciales o modificar la base de datos. El objetivo es asegurar que todas las entradas están validadas y protegidas contra ataques de inyección. Herramientas como SQLMap, Burp Suite y Havij facilitan la detección de estas vulnerabilidades.

## **PRUEBAS DE USABILIDAD**

Las pruebas de usabilidad son procedimientos destinados a evaluar la facilidad de uso, accesibilidad y satisfacción del usuario al interactuar con una aplicación. Estas pruebas aseguran que los usuarios puedan navegar y realizar tareas con facilidad, evitando confusiones o barreras de uso. Se centran en verificar la eficiencia, efectividad y satisfacción general de los usuarios. Herramientas como UsabilityHub, Hotjar, Lookback y Figma son útiles para realizar pruebas de usabilidad y obtener retroalimentación directa.

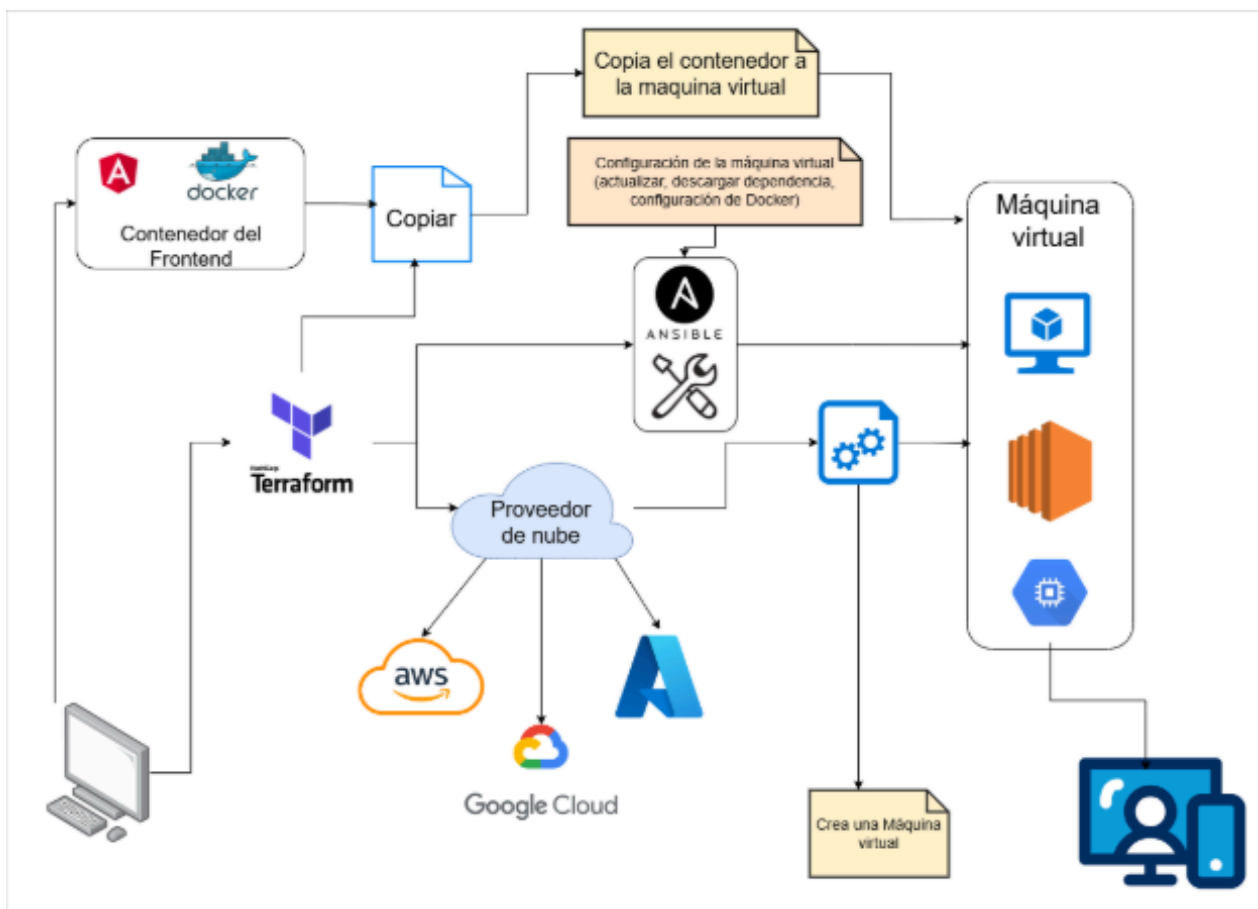
# DESPLIEGUE FRONTEND

## Uso de Terraform y Ansible para el despliegue de Frontend

Para lograr un despliegue eficiente y escalable del Frontend, utilizaremos una combinación de herramientas de Infraestructura como Código (IaC) y automatización. Implementaremos Terraform para definir y aprovisionar la infraestructura necesaria en nuestra plataforma en la nube, garantizando una configuración coherente y reproducible. Ansible se encargará de automatizar la configuración del servidor provisionado por Terraform y de orquestar el despliegue del contenedor de Docker.

Con Ansible, instalaremos y configuraremos Docker en el servidor provisionado, asegurando que esté listo para ejecutar el contenedor. Además, usaremos Ansible para desplegar el frontend de la aplicación web. Crearemos un playbook de Ansible que gestionará la construcción y el despliegue del contenedor Docker utilizando la imagen de la aplicación previamente preparada.

A continuación, se le presenta un ejemplo de cómo tendría que utilizar Terraform y Ansible.



# DESPLIEGUE BACKEND

## Uso de CI/CD

Para lograr un despliegue eficiente y escalable del Backend, se implementará un proceso de Integración Continua/Despliegue Continuo (CI/CD). Este proceso asegura que cada cambio en el código sea integrado, probado, y desplegado de manera automatizada, reduciendo errores humanos y acelerando la entrega de nuevas versiones del software. A continuación, se describen los tres estados clave que se utilizarán en este proceso:

## BUILD

En esta etapa, el código fuente se compila y se generan los artefactos necesarios para el despliegue, asegurando que el código sea válido y esté listo para ser probado.

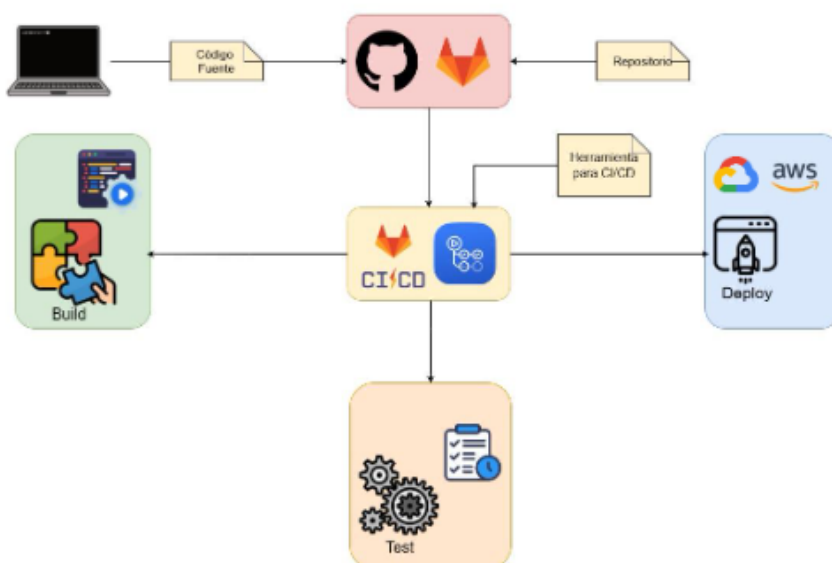
## TEST

Durante esta fase, se ejecutan pruebas automatizadas para validar la funcionalidad del código. Esto incluye pruebas unitarias y de integración, para garantizar la calidad del software.

## DEPLOY

En esta etapa final, el código que ha pasado las pruebas es desplegado en los entornos de producción o de pruebas. Este paso puede incluir el despliegue a servidores, contenedores, o servicios en la nube, garantizando que la aplicación esté disponible para los usuarios finales.

A continuación, se presenta un ejemplo de cómo sería el uso de CI/CD



La herramienta para el uso de CI/CD queda a discreción del grupo cuál utilizar (Jenkins, GitLab CI/CD, GitHub Action)

# 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 3 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 5 días por lo tanto como mínimo debe contar con 15 daily en su documentación.

Tanto para el sprint planning como para el retrospective se debe realizar una grabación donde todos los participantes estén colaborando (Es importante tomar en cuenta de las duración para cada uno).

# ENTREGAS:

## FASE I (FECHA DE ENTREGA 15 DE DICIEMBRE 2024)

1. Antecedentes
2. Core del negocio
  - a. Descripción
  - b. Diagrama de CDU de Alto nivel y su primera descomposición (CDU de alto nivel)
3. Lista de requerimientos funcionales generales (Funcionales y No Funcionales)
  - a. **Requerimientos funcionales:** Definen las funcionalidades específicas que el sistema debe realizar. Además cada requerimiento debe incluir las características asociadas (qué hará el sistema).
  - b. **Requerimientos No funcionales:** Establecen atributos de calidad como rendimiento, seguridad y escalabilidad, entre otros. Su impacto se refleja en la necesidad de optimizar algoritmos, usar tecnologías específicas o escalar los recursos hardware para cumplir con las exigencias operativas.
4. Diagrama de CDU expandidos
5. Matrices de trazabilidad
  - a. Stakeholders vrs Requerimientos
  - b. Stakeholders vrs CDU
  - c. Requerimiento vrs CDU
6. Selección del o los estilos arquitectónicos
7. Diagrama de despliegue y componentes para mostrar la arquitectura

**NOTA:** Los Frameworks se evaluarán en este punto mediante el uso de los diagramas antes mencionados, ya que al momento de realizar la documentación deben explicar cómo estos ayudan a tener un mejor entendimiento de la arquitectura del sistema.
8. Diagrama entidad relación o Esquemas detallado sobre Base de datos NoSQL
9. Los prototipos realizados para la elaboración de sus interfaces.
10. Debe realizar un mínimo de cinco patrones de diseño
11. 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

## **FASE II (FECHA DE ENTREGA 22 DE DICIEMBRE 2024)**

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

### **1. ROL CAJERO**

- a. Login
- b. Pago de servicios
- c. Pago de prestamos
- d. Buscar por Cuenta o por CUI
- e. Mostrar Saldo
- f. Generar Comprobante
- g. Retiro de Dinero
- h. Depositar Dinero
- i. Cambiar moneda (Quetzales a Dólares)
- j. pago de tarjeta de crédito

### **2. ROL ATENCIÓN AL CLIENTE**

- a. Login
- b. Creación de Cuenta (Monetario o Ahorro)
- c. Actualización de Datos del Cliente
- d. Permitir Cuentas en Dólares
- e. Encuestas de Satisfacción
- f. Registro de Quejas

### **3. ROL ADMINISTRADOR DE SISTEMAS**

- a. registro empleado
- b. asignación de roles
- c. Gestión de Copias de Seguridad

### **4. ROL SUPERVISOR**

- a. Login
- b. Encuestas de Satisfacción
- c. Ver Registro de Quejas
- d. registrar administrador de sistemas
- e. información de los administrador de sistemas
- f. Monitoreo de Actividad en Tiempo Real
- g. Gestión de Inventarios Bancarios

### **5. Actualizar documentación**

- a. Manual de usuario
- b. Actualizar documentación (Atender las observaciones y correcciones realizadas durante la fase 1).

### **6. Utilización de Docker**

### **7. Documentación Sprint**

### **8. Tablero Kanban**

### **9. Estrategia de Branching**

## **FASE III (FECHA DE ENTREGA 28 DE DICIEMBRE 2024)**

¿Qué debe realizar en esta fase del proyecto?

### **1.ROL ATENCIÓN AL CLIENTE**

- a.Crear Nuevas Tarjetas (Crédito o Débito)
- b.Bloqueo de Tarjetas
- c.Solicitudes de Cancelación de Servicios
- d.Solicitud de Préstamos al Banco

### **2.ROL ADMINISTRADOR DE SISTEMAS:**

- a.eliminar empleados
- b.cambio de contraseña empleado

### **3.SUPERVISOR**

- a.Modificar información de empleado
- b.aceptar préstamo
- c.aprobar tarjetas (Crédito o Débito)
- d.aprobar Cancelación de Servicios
- e.reportes

### **4.PRUEBAS**

- a.Pruebas unitarias
- b.Pruebas de integración
- c.Pruebas de aceptación
- d.Pruebas E2E
- e.Pruebas no funcionales

### **5. DESPLIEGUE**

- a.Uso de Terraform y Ansible
- b.Uso de CI/CD

### **6.Documentación final**

- a.Manual de usuario
- b.Documentación actualizada (Atender las observaciones y correcciones realizadas durante la fase 2)

### **7.Documentación Sprint**

### **8.Tablero Kanban**

### **9.Estrategia de Branching**

# CONSIDERACIONES:

1. Es **obligatorio** el uso de una estrategia de branching.
2. El Frameworks, así como el lenguaje de programación tanto para el desarrollo de Backend como el de Frontend queda a discreción del grupo.
3. Todo debe estar en la nube, **no se calificará nada local**.
4. Aplicar buenas prácticas para el uso de los commits.
5. Las copias parciales o totales tendrán una nota de 0 y será reportada a la escuela de sistemas.
6. Todos los integrantes deben estar presentes el día de la calificación, ya que se validará la participación de los integrantes.
7. 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.
8. Agregar al auxiliar al repositorio del grupo: GitHub -> "Mau-Noj" ; GitLab -> @brandonromero1964
9. 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.
10. Para los datos sensibles como contraseñas y tarjetas de crédito se le solicita al estudiante que cifre los datos, la herramienta queda a discreción del grupo.
11. La documentación debe ser almacenada en el repositorio utilizando el formato Markdown.
12. Almacenamiento de imágenes, deberán utilizar Google cloud storage o un bucket de Amazon S3.
13. La visualización de los archivos PDF es flexible y se puede realizar utilizando cualquier herramienta disponible.