

Prueba técnica para analista de sistema.

Duración de la Prueba: 4 horas

Objetivo: Evaluar la capacidad del candidato para analizar y diseñar un software, enfocándose en los requerimientos, el diseño del sistema, la selección de tecnologías y la presentación de soluciones innovadoras y eficientes.

Instrucciones: Se proporcionará un escenario que detalla los requerimientos para un software destinado a una casa de cambio de divisas digital. El candidato deberá analizar los requerimientos, proponer un diseño de sistema y responder a una serie de preguntas que abarcan aspectos técnicos y analíticos del diseño propuesto.

Escenario: Una empresa emergente desea desarrollar un software para su nueva casa de cambio de divisas digital. La plataforma permitirá a los usuarios comprar y vender divisas de manera rápida y segura. Entre los requerimientos clave se incluyen:

1. Interfaz de usuario amigable para realizar operaciones de cambio de divisas.
2. Integración con sistemas de pagos electrónicos y bancarios para transacciones en diferentes monedas.
3. Capacidad de actualización en tiempo real de tasas de cambio.
4. Implementación de medidas de seguridad robustas para proteger las transacciones y los datos de los usuarios.
5. Sistema de reportes para los administradores sobre las transacciones realizadas, tasas aplicadas, y tendencias del mercado.
6. Módulo de gestión de riesgos para monitorizar y alertar sobre transacciones sospechosas.

Tareas:

1. Análisis de Requerimientos:

- Identifica y prioriza los requerimientos del sistema.
- Proporciona un análisis de stakeholders involucrados.

2. Diseño del Sistema:

- Elabora un diseño preliminar del sistema que incluya arquitectura, módulos y flujo de datos.
- Describe cómo se integraría el sistema con fuentes de datos externas (por ejemplo, tasas de cambio, bancos).

3. Tecnologías y Herramientas:

- Recomienda tecnologías específicas (lenguajes de programación, bases de datos, frameworks) para el desarrollo del sistema.
- Justifica la selección de estas tecnologías.

4. Seguridad y Cumplimiento:

- Describe las estrategias para asegurar la integridad y seguridad de las transacciones y los datos de los usuarios.
- Menciona cómo el sistema podría cumplir con las regulaciones financieras relevantes.

5. Escalabilidad y Mantenimiento:

- Propone medidas para garantizar la escalabilidad y el mantenimiento a largo plazo del sistema.

Entrega: El candidato deberá presentar un informe que incluya el análisis, diseño propuesto, y respuestas a las tareas asignadas. Se valorará la claridad, profundidad y creatividad en las respuestas.

Contenido

ANÁLISIS DE REQUERIMIENTOS	3
Diseño del sistema	4
Tecnologías y herramientas	5
Seguridad y cumplimiento:	6
Escalabilidad y Mantenimiento	7
Estrategias para el mantenimiento a largo plazo	7

ANÁLISIS DE REQUERIMIENTOS

Análisis de stakeholders

Requerimientos según prioridad

Alta

- Interfaz de usuario amigable donde el sistema debe permitir que los usuarios realicen fácilmente el cambio de sus divisas
- La interfaz del usuario también debe permitir ver sus compras.
- Integración con sistemas de pago
- Seguridad y encriptado de información sensible.
- Proporcionar información en tiempo real del tipo de cambio de divisas.

Media

- Reportes de transacciones, tasas aplicadas, ganancias.
- El sistema debe cumplir con las regulaciones financieras.

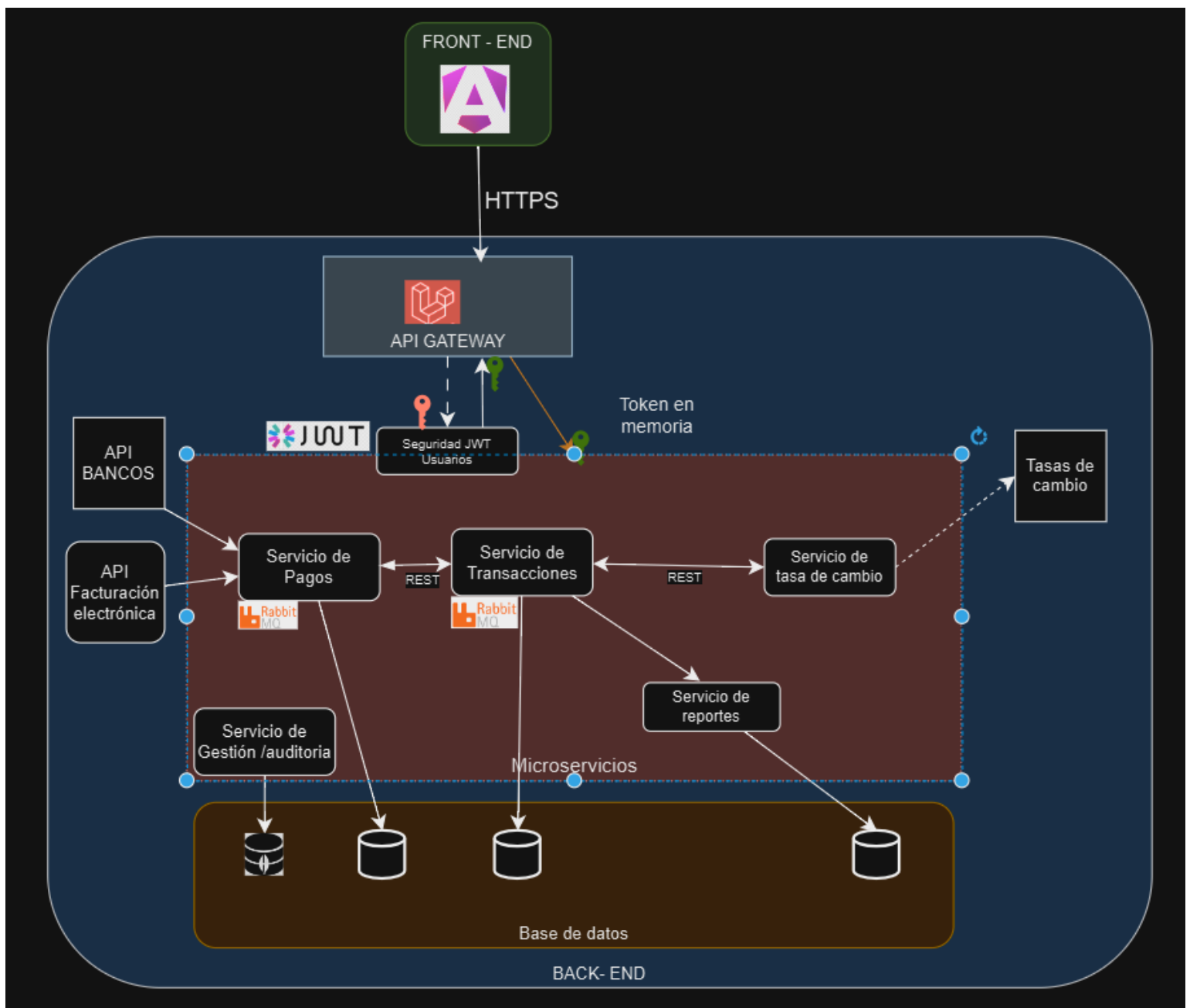
Baja

- Módulo de gestión de riesgos.
- Reportes de auditoría.

Análisis de los stakeholders

- Cliente (usuario final)
Necesitan seguridad en sus transacciones y fácil de usar.
- Administrador de sistema
Necesita herramientas para tomar decisiones (reportes).
- Desarrolladores
Necesitan tener claro los requerimientos del sistema para poder elaborar el sistema.
- Entidades financieras: Son los que revisarán los reportes o actividad sospechosa (SBS).
- Proveedores de servicios
APIs de tasa de cambio, transferencias bancarias, Sunat.
- Dueño del sistema de cambio de divisas
Interesado en el rendimiento del sistema, el vigilar lo que ocurre dentro del sistema.

Diseño del sistema



El usuario se conecta al API Gateway y entra al servicio de seguridad ahí es donde se valida si se está autenticado o si quiere autenticarse con su usuario y contraseña. En caso se autentique se crea un token el cual tendrá un tiempo de vigencia (este quedará en memoria por un determinado tiempo).

Una vez autenticado el usuario según su rol podrá acceder a los servicios que tiene permitido acceder.

El servicio de tasas de cambio se utiliza para verificar en tiempo real el tipo de cambio de las divisas, este hace una consulta a una API Rest externo como `exchangeratesapi`, `currencylayer` entre otros.

El servicio de transacciones se utiliza para iniciar una compra o venta de divisas, se valida que el usuario tenga fondos suficientes.

Si todo está bien el servicio de transacciones se comunica con el servicio de pago para procesar el pago, este se conecta a otras API bancarias para la transferencia de fondos y facturación. Luego de terminado el pago se registra en el servicio de transacciones y posterior al servicio de reportes.

El servicio de auditoria guardará información en todo momento por el uso de los servicios suscritos.

Los servicios de Pagos y transacciones usan RabbitMQ implementando colas.

Integración de Servicios externos.

Esta arquitectura está basada en microservicios en Laravel donde cada microservicio tiene conexión a su propia base de datos.

La integración con servicios externos será mediante APIs Rest para las tasas de cambio de divisa y transacciones bancarias.

Tecnologías y herramientas

Para el Font-end:

- Se utiliza Angular para el desarrollo Ágil, para realizar pruebas, para darle reactividad a la parte visual.

Para el back-end

- Cifrado SSL
- Microservicios: Se decidió trabajar con microservicios porque permite avanzar mejor al equipo de desarrollo y también para que el sistema no sea un monolito, si deja de funcionar algún servicio lo demás aún puede seguir funcionando.
- Laravel: Se utiliza para generar los microservicios rápidamente además contiene paquetes muy útiles para la elaboración de roles y permisos (como el `spatie` o `shield`).

- RabbitMQ: Para el manejo de colas otro similar es Redis, pero ambos funcionan bien de forma nativa con Laravel.
- JSON WEB TOKEN, para el método de autenticación.
- Bases de MySQL para el servicio de transacciones, seguridad y pagos
- MongoDB para el servicio de gestión de riesgos.

4. Seguridad y Cumplimiento:

- Describe las estrategias para asegurar la integridad y seguridad de las transacciones y los datos de los usuarios.
- Menciona cómo el sistema podría cumplir con las regulaciones financieras relevantes.

5. Escalabilidad y Mantenimiento:

- Propone medidas para garantizar la escalabilidad y el mantenimiento a largo plazo del sistema.

Entrega: El candidato deberá presentar un informe que incluya el análisis, diseño propuesto, y respuestas a las tareas asignadas. Se valorará la claridad, profundidad y creatividad en las respuestas.

Seguridad y cumplimiento:

Para asegurar la integridad y seguridad se utilizará lo siguiente

Autenticación con JWT y doble verificación (SMS o token por correo).

Cifrado de datos SSL/TLS,

El sistema verificará la identidad de los usuarios registrados por medio del escaneo de documentos de identidad.

Registro de auditoria de todos los servicios

El sistema cuenta con la implementación de roles y permisos

Para cumplir con las regulaciones financieras, el sistema deberá cifrar los datos sensibles de los usuarios, cumplir con auditorías y ser transparente con la SBS respecto a las transacciones que se realizan. Con la implementación de reportes de transacciones o gestión, ya se tiene lo necesario para dar evidencia de las transacciones realizadas en caso de alguna auditoría por parte de las entidades reguladoras.

Escalabilidad y Mantenimiento

Para la escalabilidad se usa una arquitectura de microservicios donde cada servicio cumple con una funcionalidad en específico, facilitando el desarrollo de forma independiente.

Se pone en memoria caché perfiles de usuario, configuración del sistema o consultas frecuentes a tablas catálogo y se indexa las tablas relacionadas con otras.

Estrategias para el mantenimiento a largo plazo

Es importante mantener una documentación actualizada en cada cambio del sistema.

Realizar pruebas unitarias para cada servicio del sistema, esto asegura que cuando se modifiquen otros servicios una verificación rápida de que está funcionando correctamente.

Implementar el load balancer y otros servicios cloud esto replicará algunos servicios.

