



**Subdirección de
Recursos Tecnológicos
-Departamento de bases de datos-**

Evaluación

Desarrollador de Sistemas de Información

Confidencial

¡Hola!

Queremos agradecerte por el tiempo invertido dentro del proceso de Selección.

Hoy te plantaremos un proyecto donde pondrás en práctica tus conocimientos y la experiencia que has obtenido en el transcurso de tu trayectoria laboral. Esta prueba está enfocada en tus conocimientos de desarrollo de software backend y frontend.

Nombre: Daniel Barrera

Correo electrónico: danielbf151100@gmail.com

Github: <https://github.com/DDanielABF>

Toda la información relacionada con código, versionamiento y documentación asociada debe ser publicada en Github, incluido este documento con las respuestas a su evaluación.

No olvide enviar el enlace de su proyecto al evaluador, dándole acceso correspondiente o dejando el acceso de forma pública.

Evaluación teórica -50pts-

Instrucciones:

A continuación, se presenta una serie de enunciados los cuales tiene como objetivo medir su capacidad de toma de decisión, así como conocer su punto de vista en el desarrollo de sistemas informáticos. Todos los enunciados de selección múltiple deben justificar su respuesta.

Parte 1: Scrum

1. ¿Qué es Scrum?

- a. Una metodología de desarrollo de software
- b. Una metodología ágil de proyectos
- c. Un marco de trabajo
- d. Otro

Justifique su respuesta: Es un marco de trabajo para metodologías ágiles de proyectos en la cual se destaca por su flexibilidad

2. ¿Cuál de los siguientes actores no deben de existir en Scrum?

- a. Scrum Master
- b. Product Owner
- c. Project Manager
- d. Developer team
- e. Stakeholder
- f. Ninguna de las anteriores

Justifique su respuesta: este actor no existe en scrum debido a que esta dividido entre el Scrum Master y el Product Owner

3. Estamos realizando el sprint 2, pero nos damos cuenta de mejoras que podemos incluir en el sprint 3, ¿lo podemos hablar en medio del sprint 2 o hasta el Sprint Planning 3?

- a. Sí, se puede hablar en el 2, es el Sprint Refinement

- b. No, eso le corresponde al Sprint Planning 3
- c. Lo decide el Scrum Master
- d. Lo decide el Product Owner

Justifique su respuesta:

4. El objetivo principal de scrum es:

- a. Entregas ágiles y continuas
- b. Desarrollo de software
- c. Hablar más con el cliente o interesado
- d. Generar valor vía el desarrollo de un producto
- e. Otro

Justifique su respuesta:

Scrum se trata de entregar el mayor valor posible al cliente. Se trabaja en ciclos cortos (sprints) en los que, paso a paso, se va construyendo y mejorando el producto. Aunque se busca mantener una comunicación constante y hacer entregas continuas, la idea central es que cada mejora o versión del producto aporte algo realmente valioso y útil para el negocio.

Parte 2: Desarrollo de software y base de datos

- 5. Es el momento de subir el código a control de cambios, usted está en la rama de desarrollo local y debe ser trasladado a producción, pasando por ambiente de pruebas. Indique cuál es la secuencia de instrucciones ideal para validar la integridad del sistema y el control de versiones.**

R//

- 1. Commit y prueba local: Realizar commits en la rama de desarrollo local asegurándose de que el código cumple con los estándares y pasa las pruebas unitarias.*
- 2. Integración en rama de relese: Fusionar los cambios en una rama de integración donde se ejecuten pruebas más amplias (integración, regresión, etc.).*
- 3. Validación en ambiente de pruebas: Desplegar el código en un ambiente de pruebas para realizar pruebas manuales y/o automáticas, validando la integridad y funcionalidad del sistema.*
- 4. Revisión y aprobación: Una vez validados, realizar una revisión de código o aprobación formal según las políticas del equipo.*

5. *Merge a producción: Fusionar la rama de integración a la rama de producción y proceder al despliegue final en el entorno productivo.*
6. *Monitoreo post-despliegue: Supervisar el sistema en producción para detectar cualquier incidencia y asegurar la estabilidad del despliegue.*

6. Una versión publicada en producción presenta inestabilidad, ¿qué acciones toma usted?

1. *Usar rama de emergencia: Crea una rama hotfix a partir de main para trabajar en la corrección.*
2. *Rollback (si es necesario): Si la situación es crítica y la corrección no está lista o causa problemas adicionales o por cuestiones de tiempo, ejecuta un rollback para restaurar la versión estable anterior.*
3. *Aislar y corregir: Reproduce el problema, identifica la causa y aplica la solución en la rama hotfix.*
4. *Pruebas exhaustivas: Valida la corrección en un ambiente de pruebas.*
5. *Despliegue controlado: Una vez comprobada la solución, realizaria merge en main y desplegaría la actualización.*
6. *Integrar los cambios de la rama hotfix en develop para mantener el control de versiones y evitar conflictos.*
7. *Retroalimentación: Documentar el incidente y analizar en la retrospectiva para mejorar el proceso.*

7. Defina con sus propias palabras qué es un modelo maestro-detalle.

R// *Un modelo maestro-detalle es, básicamente, una forma de organizar la información en dos niveles: uno principal y otro secundario. Esta estructura ayuda a mantener la información ordenada y a relacionar de forma clara los datos generales con sus componentes específicos.*

8. ¿Qué es un ORM y qué ventajas ofrece en el desarrollo backend?

R// *Es una herramienta que permite trabajar con bases de datos de forma orientada a objetos, sin tener que escribir SQL directamente. En lugar de manipular tablas y registros, lo haces a través de clases y objetos en tu lenguaje de programación.*

Ventajas:

- *Abstracción y simplificación: Facilita la interacción con la base de datos, ya que convierte las operaciones complejas en métodos y funciones fáciles de usar.*

- *Código más limpio y mantenible: Al trabajar con objetos, el código resulta más intuitivo y se integra mejor con la lógica de la aplicación.*
- *Menos errores y mayor seguridad: Al evitar escribir consultas SQL manualmente, se reduce el riesgo de errores y vulnerabilidades como inyecciones SQL.*

9. Está trabajando con manejo de documentos digitales y estos deben ser resguardados como probatorios de transacciones, es decir deben almacenarse, y estar disponibles a consulta. ¿Usaría File System, bases no relacionales u otro gestor? Desarrolle su respuesta.

R// *Para este caso, yo consideraría un enfoque combinado:*

1. *Almacenamiento de los archivos:*
Usar un sistema de almacenamiento de objetos (como Amazon S3, Google Cloud Storage o incluso un File System robusto con replicación y backups) es ideal para guardar los documentos digitales en sí. Esto te asegura durabilidad, escalabilidad y alta disponibilidad. Un simple file system local puede no ser suficiente si buscas redundancia y seguridad en ambientes productivos.
2. *Gestión y consulta de metadatos:*
Los documentos, al ser probatorios de transacciones, generalmente necesitan asociarse con información adicional (fecha, autor, tipo de transacción, etc.). Para eso, una base de datos (puede ser relacional o no relacional, dependiendo de la complejidad y flexibilidad requerida) te permitirá almacenar y consultar estos metadatos de forma eficiente.
 - *Una base de datos no relacional (como MongoDB) es ideal si tus documentos tienen estructuras variables o si necesitas mayor flexibilidad.*
 - *Una base de datos relacional puede ser más apropiada si la estructura de tus metadatos es fija y requieres transacciones fuertes.*
3. *Consideraciones adicionales:*
 - *Seguridad y auditoría: Es importante implementar mecanismos de seguridad (acceso controlado, cifrado, etc.) y mantener logs o auditorías para garantizar la integridad y la trazabilidad de cada documento.*
 - *Escalabilidad y rendimiento: La solución elegida debe poder crecer conforme aumenta el volumen de documentos sin afectar la velocidad de acceso o consulta.*
 - *Cumplimiento normativo: Si los documentos deben ser probatorios legalmente, asegúrate de que el sistema cumpla con las normativas y estándares requeridos (por ejemplo, mantener la cadena de custodia, firmas digitales, etc.).*

10. Dado los diferentes servicios provistos por la institución, en ocasiones es necesario desarrollar en diversas tecnologías y adaptar diferentes dispositivos para la recolección y lectura de datos, documentales y biométricos. Indique la estrategia que utilizaría para elegir lenguaje, tipo de desarrollo (web o desk) y las consideraciones que incluiría para selección de cada desarrollo. (Clave de servicios: Enrolamiento de pasaporte, consulta de paso migratorio, registro de extranjeros en el país, registro de refugiados, registro de retornados, consultas ciudadanas, portal web). Desarrolle su respuesta.

R//

Primero, para guardar los archivos en sí, usaría un sistema de almacenamiento de objetos, como Amazon S3 o Google Cloud Storage, o un file system robusto que tenga replicación y backups. Esto me da seguridad, escalabilidad y disponibilidad, ya que el archivo se guarda de manera que no se pierda y pueda crecer conforme se agregan más documentos.

Segundo, como estos documentos suelen tener información adicional (por ejemplo, fecha, autor, tipo de transacción), también considero importante guardarlos en una base de datos que administre esos metadatos. Dependiendo de lo que necesite, podría usar:

- *Una base de datos no relacional, como MongoDB, si la estructura de los datos es variable y necesito más flexibilidad.*
- *Una base de datos relacional, si los datos tienen una estructura fija y requiero transacciones fuertes.*

Además, es fundamental asegurar que el sistema cumpla con los requisitos de seguridad y normativas legales, lo que implica tener control de acceso, cifrado y registros de auditoría para garantizar la integridad y la trazabilidad de cada documento.

Evaluación práctica -50pts-

Instrucciones

Esta sección tiene como objetivo medir tu nivel de conocimiento técnico en el área de desarrollo de sistemas de información. A continuación, se te brinda una serie de requerimientos funcionales con los cuales debes construir un sistema básico de registro de usuarios en un sistema.

1. Se debe crear una base de datos que maneje usuarios de un sistema. Esto debe incluir nombre, correo electrónico, contraseña como mínimo, así como tipo (administrador o ciudadano), tomando en consideración que un ciudadano puede ser un administrador.
2. El usuario contará con un id único de identificación el cual será utilizado como id de login.
3. Realizar el formulario de registro y login del sistema para usuarios.
4. Debe crearse una tabla de datos de pasaportes, los cuales deben incluir como mínimo tipo de pasaporte, fecha de emisión, fecha de vencimiento, lugar y país de emisión, número de pasaporte, asociado a los datos del usuario.
5. Cada usuario registrado en el sistema puede tener asignado 1 o más pasaportes, pero solo uno de ellos puede estar activo a la vez. Un pasaporte solo puede estar asociado a un usuario.
6. El usuario puede modificar sus datos y / o actualizar su contraseña, mas no puede modificar su id único. No puede modificar datos de pasaporte, únicamente datos personales.
7. Debe existir un usuario administrador, el cual pueda ver una lista de todos los usuarios y de todos los pasaportes registrados, con opción a ver detalle en cada caso.
8. El administrador puede habilitar o deshabilitar pasaporte, así también puede asociar un nuevo pasaporte al usuario.
9. El administrador puede deshabilitar un usuario dentro del sistema.

Para la realización de esta tarea deberá realizar:

1. Base de datos de las tablas necesarias, incluir esquema entidad-relación y tabla de metadatos
2. Api de servicios para manejo de datos entre frontend y backend
3. Front end para la visualización de lo solicitado (puede utilizar la interfaz que considere oportuna)

Aspectos que evaluar

- Diseño general
- Manejo de errores

- Usabilidad
- Interactividad

Lenguajes:

- Node.JS
- C#
- Api Rest
- React /Vue
- MySQL/PostgreSQL/SQLServer