



**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:Erick Enrique González Chávez

Correo electrónico: [gcerickenrique@gmail.com](mailto:gcerickenrique@gmail.com) /  
[gonzalezerickenrique21@gmail.com](mailto:gonzalezerickenrique21@gmail.com)

Github:

<https://github.com/Erick21Enrique99Gon/RegistroDeUsuario>

---

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:

La metodología scrum es una metodología ágil de desarrollo que permite una entrega continua, una revisión de avances por partes, asignación de tareas que se deben de cumplir en tiempo y forma correcta.

### 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:

Deben existir todos los actores involucrados ya que cada uno contribuya al desarrollo del proyecto dejar a uno de ellos afuera afectará el desarrollo del proyecto, si se pierde el objetivo del proyecto o si se dejan de tener los recursos de mano de obra.

**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:

El scrum master debe de analizar el estado de los sprints ya que el agregar más tareas puede crear sobrecarga de trabajo o si las mejores lo ameritan pueden sólo cambiar de objetivo y finalidad ahorrando así tiempos de desarrollo.

**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:

El desarrollo de scrum se realizan sprints en los cuales se dividen responsabilidades y las tareas dejando tiempos para ser completadas y de esta manera realizar avances continuos en el desarrollo.

## 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.**

Se inicia con un commit de los cambios a la rama feature y después se realiza pull request a la rama developer o la rama que se tenga como quality assurance, en la cual se deben tener pruebas unitarias, se deben realizar la pruebas de integración para comprobar el comportamiento de los componentes, se debe actualizar al equipo de quality assurance para realicen las pruebas necesarias, y al comprobarse totalmente su comportamiento se realiza el merge a la rama de producción.

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

Se deben de realizar una documentación del problema para poder ser analizado, al poder encontrar el problema exacto se puede realizar un hotfix documentado para realizar auditorías y mejoras.

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

El modelo se trata del control de la información aumentando el detalle en el momento en que se baja de la jerarquía.

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

ORM ofrece un manejo de la base de datos de manera que se establece un espejo de la estructura de la base en backend evitando así la realización de procedimientos en el motor de base de datos que se seleccionó, evitando un desacople de la información y la metadata.

**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.**

Seleccionaría una base de datos no relacional, esto me permitirá el manejo eficiente de las búsquedas, teniendo una metadata que permite la búsqueda de la información así como facilitar el estado en el que se encuentra.

**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.**

Se realiza una toma de requerimientos funcionales y no funcionales, y las restricciones esto nos ayudará a realizar un planteamiento de los distintos servicios.

Se realizan los casos de uso de cada servicio para poder realizar una buena comunicación y plantear el entendimiento que se tiene de los servicios.

Se realizan las matrices de trazabilidad lo que nos permite establecer a que stakeholder se debe de dirigir para obtener retroalimentación y caso de uso se tiene por requerimiento funcional permitiendo observar que como se plantearon en su uso.

Con esta información ya se puede empezar a tomar decisiones técnicas como es el diagrama de componentes, este diagrama nos permite establecer partes generales del código, su agrupación y su posible reutilización, estos sembraran las decisiones arquitectónicas del software.

Este diagrama nos ayudará a realizar los diagramas de paquetes agrupando componentes, los paquetes a su vez nos ayudarán a plantear la arquitectura de despliegue.

En el diagrama arquitectura de despliegue se tendrá toda la información y análisis para poder plantear diversas soluciones, en este diagrama se plantea si será un service desk o se podrá conectar a web.

Al tener todos estos elementos se puede realizar la elección de lenguaje, frameworks y motores para el desarrollo de los proyectos.

Estas decisiones técnicas nos ayudan a plantear decisiones de gran impacto en el desarrollo como es el diagrama de bloques que es un diagrama de alto nivel que nos ayudará a plantear soluciones y exponerlas a los stakeholders, con este diagrama se tratara de explicar explicar los recursos necesarios para el desarrollo y despliegue a personas que no tenga conocimientos técnicos.

# 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