



SI653 - APLICACIONES WEB
EXAMEN FINAL
2022-1

Sección: ES3B, ES3C, WX51
Profesores: Velásquez Núñez, Ángel Augusto

Duración: 170 minutos

Indicaciones:

1. El examen consta de 1 pregunta, y tendrá 170 minutos para resolverlas.
 2. La pregunta es de tipo Proyecto de Software y la entrega de su respuesta es a través de envío de archivo empaquetado **.zip** con nombre *upc-pre-202201-si653-<sección>-eb-u<código-estudiante>.zip*, conteniendo el proyecto de software, en la Actividad para el Examen final.
 3. Cada examen cuenta con un equipo académico, el cual estará conectado durante los primeros **15 minutos del examen**.
 4. El alumno debe dedicar los primeros 15 minutos a revisar las preguntas del examen y de presentarse alguna duda enviar un correo al(los) profesor(es)
Velásquez Núñez, Ángel Augusto, correo pcisavel@upc.edu.pe
 5. De no recibir respuesta del equipo académico, o tener algún inconveniente adicional pasado los primeros 15 minutos, puede comunicarse con el profesor
Delgado Vite, Jorge Luis, correo pcsijdev@upc.edu.pe
 6. Los profesores en mención, solo recibirán correos provenientes de las cuentas **UPC**, de ninguna manera se recibirán correos de cuentas públicas.
 7. Ante problemas técnicos, debe de forma obligatoria adjuntar **evidencias** del mismo, como capturas de pantalla, videos, fotos, etc. Siendo **requisito fundamental** que, en cada evidencia se pueda apreciar claramente la **fecha y hora del sistema operativo del computador** donde el alumno está rindiendo el examen.
 8. Los problemas técnicos se recibirán como máximo 15 minutos culminado el examen.
-

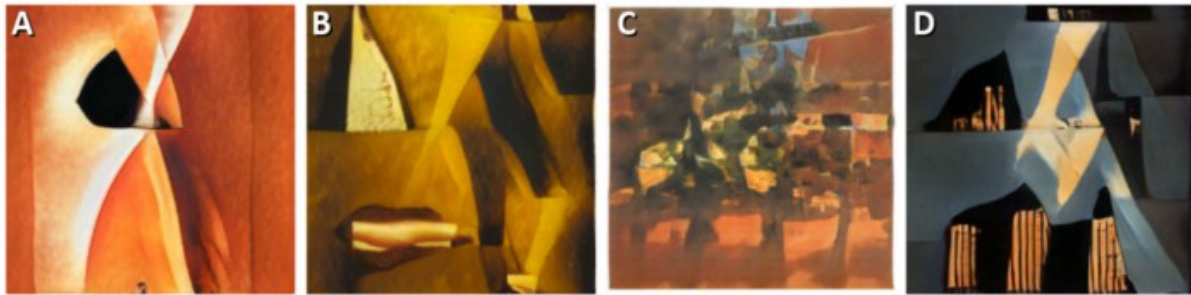
Enunciado:

Caso magnet.art

Robert A. Gonsalves es un artista, inventor e ingeniero en Boston quien investiga y escribe sobre el uso creativo de la inteligencia artificial (Artificial Intelligence o AI).

El proyecto más reciente en el que ha estado trabajando se llama MAGnet. Robert construyó un custom Genetic Algorithm (GA) para dirigir la creación de pinturas modernas utilizando Generative Adversarial Network (GAN) a partir de una descripción en texto sobre varias generaciones. MAGnet utiliza el modelo CLIP de OpenAI y una variante de StyleGAN2 ADA de Nvidia llamada SWAGAN, que utiliza wavelets para crear imágenes.

MAGnet permite, a partir de un criterio especificado en texto como “an abstract painting with orange triangles”, consultar un banco de imágenes y en primera instancia y obtener un conjunto de imágenes que cumplen el criterio, como por ejemplo:



Luego de ello, MAGnet aplica un proceso de ejecución del GA por cinco generaciones, llegando a un resultado creativo final:



El sistema MAGnet funciona bastante bien. Parece realizar un mejor trabajo generando pinturas abstractas que pinturas figurativas. Converge en una solución bastante rápido, en cinco generaciones.

Es por ello que Robert decide ofrecer a la comunidad de usuarios la posibilidad de generar obras de arte generadas por AI, a través del website magnet.art.

Para poder realizar el proceso de generación, se requiere la construcción de datasets para el proceso de entrenamiento. Para ello se integra con el API de WikiArt.org, aunque se puede incorporar otros proveedores de imágenes para diversificar la variedad de estas.

Pregunta 1 (20 p.).

Usted se integra al backend software developer team, a cargo de la creación de un **RESTful API** que brinde soporte a las operaciones de magnet.art.

El ecosistema de magnet.art requiere que la aplicación *Magnet.art website* cuente con **Endpoints** en el RESTful API, para el manejo de la información de autores (**authors**) y proveedores (**providers**).

Como reglas de negocio, magnet.art:

- No permite que se registre dos **authors** con la misma combinación de **firstName** y **lastName**.
- No permite que se registre dos **authors** con el mismo **nickname**.
- No permite que existan registrados dos **providers** con el mismo **apiUrl**.
- No permite que se registre dos **providers** con el mismo **name**.
- No permite que se registre un **provider** cuando **keyRequired** es **true** y **apiKey** es vacío, blanco o nulo.
- Establece que la información que se desea preservar de los **Authors** incluye **id** (int, obligatorio, autogenerado, llave primaria), **firstName** (string, obligatorio), **lastName** (string, obligatorio), **nickname** (string, obligatorio), **photoUrl** (string, no obligatorio).
- Establece que la información de los **Providers** incluye **id** (int, obligatorio, autogenerado, llave primaria), **name** (string, obligatorio), **apiUrl** (string, no obligatorio), **keyRequired** (boolean, no obligatorio, pero false por defecto), **apiKey** (string, no obligatorio).
- En el caso de responses que retorna el API con información de **authors**, para cada **Author** se debe incluir **id**, **firstName**, **lastName**, **nickname**, pero **no** debe incluirse el **photoUrl**.

Nota: Authors y Providers son conceptos independientes, no existe relación entre ellos.

Durante la etapa de desarrollo, le asignan trabajar en específico sobre dos Endpoints:

/api/v1/authors
/api/v1/providers

Authors Endpoint (/api/v1/authors)

Debe implementar **solo dos** operaciones en el RESTful API: agregar un **author** y consultar **todos**. Los valores de **id** son autogenerados al momento de almacenar la información.

Author Providers Endpoint (/api/v1/providers)

Debe implementar **solo dos** operaciones sobre los **providers** en el RESTful API: agregar un **provider** y consultar por **id**. Los valores de **id** son autogenerados al momento de almacenar la información.

Incluya como parte del desarrollo la implementación de las reglas de negocio.

Technical constraints

1. Elabore la solución con C#, NET 6 y ASP.NET Core Framework.

2. Cree su solución y el proyecto de software con el nombre ***si653ebu<código-estudiante>.API*** (por ejemplo, *si653ebu201621873.API*).
3. Considere que el concepto **Author** pertenece al bounded context **painting**.
4. Considere que el concepto **Provider** pertenece al bounded context **training**.
5. La información debe ser persistente en una base de datos relacional (MySQL), en un esquema **ma**.
6. Aplique buenas prácticas de Arquitectura de Software, enfoque Domain-Driven, principios y patrones de diseño, convenciones de nomenclatura en inglés, así como buenas prácticas de nomenclatura en C# (entre ellas Upper Camel Case para Clases, Properties, Métodos; Lower Camel Case para atributos privados) y buenas prácticas para nomenclatura de objetos de Base de Datos (entre ellas snake case, tablas en plural, sin mnemónicos, id como nombre de primary key).
7. El puerto de escucha del API en **localhost** debe ser el puerto **7070**.
8. Utilice minúsculas para los nombres de URL para todos los endpoints.
9. Utilice Properties para representar los atributos de las clases Entity.
10. Utilice la biblioteca AutoMapper para el Object Mapping.
11. Incluya documentación de Endpoints con OpenAPI.
12. Considere la gestión de excepciones en la aplicación.
13. Empaquete su solución como un archivo **.zip**. (único formato válido) con el nombre ***upc-pre-202201-si653-<sección>-eb-u<código-estudiante>.zip*** (por ejemplo, *upc-pre-202201-si653-wx51-eb-u201621873.zip*).
14. Suba su archivo de solución en la Actividad indicada para el Examen final.

NO forma parte del alcance del proyecto:

1. Soporte de CORS.
2. Security.
3. Testing.

Rúbrica de calificación

Criterio de Calificación	Excelente	Promedio	Deficiente	Calificación
C01. Building y ejecución	Al abrir el proyecto y ordenar la ejecución, ésta se inicia sin problemas. El API es accesible en la ruta indicada.	La aplicación no llega a iniciar y ejecutarse, sin embargo el proceso de building llega a concluir.	Al cargar el proyecto el proceso de building presenta errores y no llega a concluir.	
	2.0 puntos	1.0 punto	0 puntos	
C02. Authors Endpoint	El RESTful API expone el endpoint <code>/api/v1/authors</code> . Se evidencia la funcionalidad de las operaciones CRUD Create y Get All, según las convenciones REST, proporcionando en cada caso los valores esperados y respondiendo adecuadamente ante las excepciones. Se evidencia la persistencia de los objetos Author, cumpliendo estructura según enunciado, en una tabla <i>authors</i> en base de datos relacional indicada en un esquema <i>ma</i> . El Endpoint cuenta con documentación basada en OpenAPI.	El RESTful API expone el endpoints <code>/api/v1/authors</code> . Se evidencia parcialmente la funcionalidad de las operaciones CRUD Create y/o Get All según las convenciones REST, proporcionando en algunos casos los valores esperados y respondiendo de forma parcialmente adecuada ante las excepciones, o se evidencia parcialmente la persistencia de los objetos Author con estructura según enunciado, en una tabla <i>authors</i> en base de datos relacional indicada en un esquema <i>ma</i> .	La aplicación no implementa o expone el endpoint <code>/api/v1/authors</code> .	
	4.0 puntos	2.5 puntos	0 puntos	
C03. Providers Endpoint	El RESTful API expone el endpoint <code>/api/v1/providers</code> . Se evidencia la funcionalidad de las operaciones CRUD Create y Get by id, según las convenciones REST, proporcionando en cada caso los valores esperados y respondiendo adecuadamente ante las excepciones. Se evidencia la persistencia de los objetos Provider, cumpliendo estructura según enunciado, en una tabla <i>providers</i> en base de datos relacional indicada en un esquema <i>ma</i> . El Endpoint cuenta con documentación basada en OpenAPI.	El RESTful API expone el endpoint <code>/api/providers</code> . Se evidencia parcialmente la funcionalidad de las operaciones CRUD solicitadas según las convenciones REST, proporcionando en algunos casos los valores esperados y respondiendo de forma parcialmente adecuada ante las excepciones, o se evidencia parcialmente la persistencia de los objetos Provider con estructura según enunciado, en una tabla <i>providers</i> en base de datos relacional indicada en un esquema <i>ma</i> .	La aplicación no implementa o expone el endpoint <code>/api/v1/providers</code> .	
	4.0 puntos	2.5 puntos	0 puntos	
C04. Business Rules	El desarrollo incluye la implementación de reglas de negocio, cubriendo de forma completa las condiciones y escenarios establecidos, siendo éstas ejecutables, con adecuado manejo de excepciones, implementando éstas en las capas más adecuadas, aplicando convenciones y buenas prácticas.	El desarrollo incluye la implementación de la mayoría de reglas de negocio, cubriendo de forma parcial las condiciones y escenarios establecidos, siendo éstas ejecutables, implementándolas en las capas adecuadas en la mayoría de casos, ó aplicando parcialmente convenciones y buenas prácticas.	No implementa reglas de negocio, o no cubre escenarios más allá de operaciones CRUD básicas, o éstas no son ejecutables.	
	4.0 puntos	2.5 puntos	0 puntos	
C05. Code Organization	El desarrollador organiza el código y los elementos de backend de la solución, aplicando buenas prácticas de C#, .NET Framework Core, ASP.NET Core y Domain-Driven Design, agrupando los elementos de la solución según convenciones, manteniendo organización de paquetes y carpetas recomendadas por el fabricante y buenas prácticas de la industria de software.	El desarrollador aplica parcialmente en backend convenciones, recomendaciones y buenas prácticas de C#, .NET Framework Core, ASP.NET Core y Domain-Driven Design.	No se evidencia un criterio de organización para los elementos de la solución.	
	2.0 punto	1.25 puntos	0 puntos	
C06. Code Quality	Utiliza para el backend el lenguaje de programación C#. La codificación tiene un estilo claro, indentando los bloques de código según los estándares de programación correspondientes al lenguaje, aplicando una lógica consistente en los métodos, condicionales sin escenarios no contemplados, uso adecuado de reutilización de código para evitar redundancia. Aplica patrones de arquitectura, principios y patrones de diseño. Distribuye el código en los niveles correspondientes, asignando lógica de persistencia, lógica de negocio, lógica de control, lógica de mapping y transferencia a las interfaces y clases que corresponden. Cumple de forma completa con los technical constraints.	Utiliza para el backend el lenguaje de programación C#. La codificación es funcional, pero sólo aplica parcialmente los estándares de indentación de bloques de código, ó existen ineficiencias en la codificación: redundancia ó inconsistencias en la lógica de programación. Aplica parcialmente patrones de arquitectura y patrones de diseño, o existe en algunas partes una distribución de la lógica en los niveles incorrectos. Cumple de forma parcial con los technical constraints.	No utiliza el lenguaje de programación C# para el backend, ó la codificación es funcional pero no se evidencia aplicación de estándares ó criterios de eficiencia en la codificación, con ausencia de comentarios, ó no aplica patrones de arquitectura ni patrones de diseño, o la codificación no es funcional.	
	3.0 puntos	1.75 punto	0 puntos	
C07. Naming Standards	El desarrollador aplica en todos los nombres de objetos de programación como namespaces, componentes, interfaces, clases, objetos, variables, constantes y métodos la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, así como los recursos.	El desarrollador aplica sólo en parte la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, así como los recursos.	El desarrollador no aplica nomenclatura en inglés para los objetos de programación ó recursos.	
	1.0 puntos	0.5 punto	0 puntos	
Total	20 puntos	12 puntos	0 puntos	

Lima, 07 de Julio del 2022

Anexos

Anexo A. Referencias

Comprimir y descomprimir archivos: <https://support.microsoft.com/es-es/windows/comprimir-y-descomprimir-archivos-8d28fa72-f2f9-712f-67df-f80cf89fd4e5>

REST API Tutorial: <https://restfulapi.net/>

Swashbuckle.AspNetCore - OpenAPI Library for ASP.NET Core:
<https://github.com/domaindrivendev/Swashbuckle.AspNetCore>

AutoMapper Documentation: <https://docs.automapper.org/en/latest/index.html>

MySQL Connector/NET for Entity Framework – Entity Framework Core Support:
<https://dev.mysql.com/doc/connector-net/en/connector-net-entityframework-core.html>