SI730 - APLICACIONES WEB
EXAMEN FINAL
2023-1

Sección: SW51, SW52, SW53, SV54, WS51, WS52
Profesores: Reupo-Musayón Gastulo, Naldo
Tinoco Licas, Juan Carlos
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-202301-si730-<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 el 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)

   Secciones SV54, WS52: Bautista Fuentes, Iván Christian al correo pcsiibau@upc.edu.pe
   Secciones SW53, WS51: García Rojas, Fidel Eugenio al correo pcsifgar@upc.edu.pe
   Secciones SW51, SW52: Morales Arévalo, Juan Carlos al correo pcsijumo@upc.edu.pe

5. Los profesores en mención, solo recibirán correos provenientes de las cuentas UPC, de ninguna manera se recibirán correos de cuentas públicas.
6. 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.
7. Los problemas técnicos se recibirán como máximo 15 minutos culminado el examen.

Enunciado:

Caso ISA.

En el corazón de toda cocina profesional se encuentra el frigorífico. Cualquier dueño de negocio que dependa de alimentos frescos será muy consciente de la importancia de la refrigeración comercial para su negocio. Para tales empresas, los alimentos frescos son la principal fuente de ingresos, por lo que es crucial que sus aparatos de refrigeración comercial funcionen de manera confiable y efectiva.

Mantener temperaturas óptimas garantiza que los dueños de negocios puedan ofrecer a sus clientes alimentos frescos. Si la temperatura llega a estar por encima o por debajo del rango óptimo, se producirá el deterioro y la descomposición de los alimentos. Los productos alimenticios en mal estado, el incumplimiento de las normas alimentarias y la avería de los equipos contribuyen a la pérdida de ingresos, al daño a la reputación y, a menudo, generan gastos adicionales, incluso multas.

Desde 1963 ISA (https://www.isaitaly.com/) produce en su región y desde allí provee vitrinas refrigeradas y muebles refrigerados para lugares públicos a nivel mundial. ISA actúa en el mercado a través de tres marcas: ISA, COF, TASSELLI e HIZONE con un volumen de ventas superior a los 120 millones de euros exportando, en 107 países, productos de calidad con un alto índice de tecnología e innovación, también en términos de sostenibilidad mediante el uso refrigerantes naturales.

ISA es hoy en día uno de los jugadores más importantes del mundo en el campo del diseño de vitrinas y armarios frigoríficos para heladería y pastelería. Más allá de numerosos y prestigiosos clientes, que hacen uso diario de los equipos de ISA, la empresa colabora desde hace años con importantes marcas internacionales, como: Ahold, Auchan, Autogrill, Billa, Bindi, Coldstone Creamery, Coca Cola, Conad, Coop, Cremonini , Brioche Doree, Haagen Dazs, Nestlé, Sammontana, Sturbucks y Unilever.

ISA opera en un mercado competitivo donde los clientes exigen un producto confiable y características adicionales que se suman a la experiencia del usuario. Para ISA, la conectividad IoT proporciona la base para un conjunto más amplio de nuevas características y beneficios. A continuación, exploramos algunos de los beneficios clave que IoT genera para ISA.

ISA Connect es una plataforma que incluye un conjunto de productos relacionados para facilitar el mantenimiento predictivo y el monitoreo continuo.

Para ISA, es importante que sus soluciones ofrezcan:

- Monitoreo remoto del funcionamiento de la vitrina 24/7 durante todo el año.
- Mantenimiento remoto de las funciones principales de la vitrina.
- Estadísticas operativas y análisis de datos.

Locales integrados y conectados: La tecnología ISA Connect se aplica a la amplia gama de productos ISA, desde heladerías hasta supermercados, desde los escaparates hasta las cocinas comerciales.

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 ISA.

El ecosistema de ISA Platform requiere que el ISA Connect Web Application, el Embedded IoT System y el ISA Connect Mobile Application cuenten con **Endpoints** en el RESTful API, para el manejo de la información de:

Products, conformadas por los atributos id (int, Primary Key, Autogenerado), brand (string, Obligatorio), model (string, Obligatorio), serialNumber (string, Obligatorio), status (int, Obligatorio, Valores posibles 1 = Operational, 2 = Unoperational). Adicionalmente contiene un atributo que **no** debe convertirse en columna de tabla llamado statusDescription, de tipo string, siendo sus únicos posibles valores: "OPERATIONAL", "UNOPERATIONAL" (*Tip*: Ver sección "*Included and excluded properties*" en documentación sobre "*Entity Properties*" referenciada en el Anexo A). El valor que se asigne a *statusDescription* debe afectar el valor de *status*. El valor que se lea del atributo *statusDescription* debe derivarse del valor que tenga *status*.

Maintenance Activities, conformadas por los atributos id (int, Primary Key, Autogenerado), productSerialNumber (string, Obligatorio, Serial Number del product sobre el que se realiza el activity), summary (string, obligatorio), description (string, no obligatorio), activityResult (int, Obligatorio, Valores posibles 0 = Product still Unoperational, 1 = Product is Operational).

Como reglas de negocio, ISA:

- No permite que se registre dos **products** con el mismo valor de *serialNumber*.
- No permite que se registre un **maintenance activitiy** cuyo valor de *productSerialNumber* no coincida con el valor de *serialNumber* para alguno de los *products* existentes almacenados.
- Establece que la información que se desea preservar de los ***Products,*** incluye **id** (int, Primary Key, Autogenerado), **brand** (string, Obligatorio), **model** (string, Obligatorio), **serialNumber** (string, Obligatorio), status (int, Obligatorio, Valores posibles 1 = Operational, 2 = Unoperational)
- Establece que la información de los ***Maintenance Activities***, incluye **id** (int, Primary Key, Autogenerado), **productSerialNumber** (string, Obligatorio), **summary** (string, obligatorio), **description** (string, no obligatorio), **activityResult** (int, Obligatorio, Valores posibles 0 = Product still Unoperational, 1 = Product is Operational).
- Para **Product**, al momento de ingresos o actualizaciones vía el Endpoint, el valor de **statusDescription** se debe recibir en el request como **string**, siendo los únicos posibles valores: "OPERATIONAL", "UNOPERATIONAL". Ello internamente debe afectar el valor de *status* (el cual no participa en el request).
- Para **Product**, cuando se responde una consulta, en el response, para el caso del valor de **statusDescription,** se debe retornar en el response su valor como string según corresponda, sea "OPERATIONAL" o "UNOPERATIONAL". No se incluye *status* como parte del response. Solo se incluye en el response id, brand, model
- Cuando se agrega un **Maintenance Activity**, el valor que tenga **activityResult** debe determinar si se modifica el valor de **status** de ***Product***, para que tenga coherencia con el resultado del mantenimiento.

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

/api/v1/products
/api/v1/maintenance-activities/

Products Endpoint (https://localhost:7070/api/v1/products)
Debe implementar **solo dos** operaciones en el RESTful API: agregar (POST) un **product** y consultar (GET) un product por id. Los valores de **id** son autogenerados al momento de almacenar la información. Al agregar se debe retornar en el response el objeto incluyendo el id generado para el mismo.

Maintenance Acitivities Endpoint (https://localhost:7070/api/v1/maintenance-activities)
Debe implementar **solo una** operación sobre los **maintenance activities** de score en el RESTful API: agregar (POST) un **maintenance activity**. Al agregar se debe retornar en el response el objeto incluyendo el id generado para el mismo.

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

Technical constraints

1. Elabore la solución con C#, NET 7 y ASP.NET Core Framework.
2. Cree su solución y el proyecto de software con el nombre ***si730ebu<código-estudiante>.API*** (por ejemplo, *si730ebu201621873.API*).
3. Considere que el concepto **Product** pertenece al bounded context **inventory**.
4. Considere que el concepto **Maintenance Activity** pertenece al bounded context **maintenance**.
5. Considere el bounded context **shared** con los elementos que correspondan.
6. La información debe ser persistente en una base de datos relacional (MySQL), en un esquema **isa**.
7. 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).
8. El puerto de escucha del API en **localhost** debe ser el puerto **7070**.
9. Utilice minúsculas para los nombres de URL para todos los endpoints.
10. Utilice Properties para representar los atributos de las clases Entity.
11. Utilice la biblioteca AutoMapper para el Object Mapping.
12. Incluya documentación de Endpoints con OpenAPI.
13. Considere la gestión de excepciones en la aplicación.
14. Empaquete su solución como un archivo **.zip**. (único formato válido) con el nombre ***upc-pre-202301-si730-<sección>-eb-u<código-estudiante>.zip*** (por ejemplo, *upc-pre-202301-si730-wx51-eb-u201621873.zip*).
15. 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	Sobresaliente (S)	Esperado (E)	Necesita Mejorar (M)	Insuficiente (I)	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.	No elabora solución	
	2.0 puntos	**1.0 punto**	**0.5 puntos**	**0 puntos**	
C02. Products Endpoint	El RESTful API expone el endpoint especificado en el enunciado. Se evidencia la funcionalidad de las operaciones solicitadas, proporcionando en cada caso los valores esperados y respondiendo adecuadamente ante las excepciones. Se evidencia la persistencia de los objetos solicitados, cumpliendo la estructura según enunciado, en una tabla nombrada según especificaciones, en base de datos relacional según enunciado en el esquema indicado. Se incluye documentación del Endpoint con OpenAPI.	El RESTful API expone el endpoint especificado en el enunciado. Se evidencia parcialmente la funcionalidad de las operaciones solicitadas, proporcionando en algunos casos los valores esperados y respondiendo de forma parcialmente adecuada ante las excepciones, o se evidencia parcialmente la persistencia en base de datos relacional, o se evidencia parcialmente la persistencia de los objetos solicitados con estructura según enunciado, en una tabla *nombrada según especificaciones* en base de datos relacional según enunciado en el esquema *indicado*.	La aplicación implementa y expone el endpoint especificado en el enunciado, pero no cumple con la ruta especificada o no se puede registrar elementos.	La aplicación no implementa o expone el endpoint solicitado.	
	4.0 puntos	**3.0 puntos**	**1.5 puntos**	**0 puntos**	
C03. Maintenance Activities Endpoint	El RESTful API expone el endpoint especificado en el enunciado. Se evidencia la funcionalidad de las operaciones solicitadas, proporcionando en cada caso los valores esperados y respondiendo adecuadamente ante las excepciones. Se evidencia la persistencia de los objetos solicitados, cumpliendo la estructura según enunciado, en una tabla nombrada según especificaciones, en base de datos relacional según enunciado en el esquema indicado. Se incluye documentación del Endpoint con OpenAPI.	El RESTful API expone el endpoint especificado en el enunciado. Se evidencia parcialmente la funcionalidad de las operaciones solicitadas, proporcionando en algunos casos los valores esperados y respondiendo de forma parcialmente adecuada ante las excepciones, o se evidencia parcialmente la persistencia en base de datos relacional, o se evidencia parcialmente la persistencia de los objetos solicitados con estructura según enunciado, en una tabla *nombrada según especificaciones* en base de datos relacional según enunciado en el esquema *indicado*.	La aplicación implementa y expone el endpoint especificado en el enunciado, pero no cumple con la ruta especificada o no se puede registrar elementos.	La aplicación no implementa o expone el endpoint solicitado.	
	4.0 puntos	**3.0 puntos**	**1.5 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.	El desarrollo incluye la implementación de algunas de las reglas de negocio, incumpliendo la mayoría de condiciones y escenarios establecidos, siendo éstas ejecutables, implementándolas en las capas adecuadas en muy pocos casos, ó con poca evidencia de aplicar 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	**3.0 puntos**	**1.5 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 en la mayoría de casos para el backend convenciones, recomendaciones y buenas prácticas de C#, .NET Framework Core, ASP.NET Core y Domain-Driven Design.	El desarrollador en algunos casos para el 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	**1.0 puntos**	**0.5 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, aplica en la mayoría de casos los estándares de indentación de bloques de código, ó existen algunas 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 con la mayoría de los technical constraints.	Utiliza para el backend el lenguaje de programación C#. La codificación es funcional, pero sólo aplica algunos de los estándares de indentación de bloques de código, ó existen muchas ineficiencias en la codificación: redundancia ó inconsistencias en la lógica de programación. Aplica algunos patrones de arquitectura y patrones de diseño, o existe en muchos casos una distribución de la lógica en los niveles incorrectos. Cumple de con solo algunos de 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	**2.0 punto**	**1.0 puntos**	**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 en la mayoría de casos la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, así como los recursos.	El desarrollador aplica en muy pocos casos 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**	**0.25 puntos**	**0 puntos**	
Total	**20 puntos**	**13.5 puntos**	**6.75 puntos**	**0 puntos**	

Lima, 8 de Julio del 2023

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

Properties
https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/properties

Entity Properties
https://learn.microsoft.com/en-us/ef/core/modeling/entity-properties?tabs=fluent-api%2Cwithout-nrt