



Upc-pre-202510-1asi0730-examen-final-4395 v1

Aplicaciones Web (Universidad Peruana de Ciencias Aplicadas)



Scan to open on Studocu



**1ASI0730 - APPLICACIONES WEB
EXAMEN FINAL
202510**

NRC: 4395

Profesores: Mori Paiva, Hugo Allan
Reupo-Musayón Gastulo, Naldo
Sánchez Ponce, Alex Humberto
Sánchez Seña, Alberto Wilmer
Velásquez Núñez, Ángel Augusto
Villafuerte Bazán, Óscar Iván

Duración: 170 minutos

Indicaciones:

1. El examen consta de 1 pregunta, y tendrá **170 minutos** para resolver y entregar.
 2. La pregunta es de tipo Proyecto de Software y la entrega de su respuesta es a través de envío de archivo empaquetado **.zip** (único formato válido) con nombre *eb<NRC>u< código-estudiante>.zip* (por ejemplo, eb4395u201621873.zip), conteniendo el proyecto de software, en la Actividad para el Examen final.
 3. Puede utilizar como referencia los materiales publicados en el aula virtual, los sitios referenciados en el enunciado, así como sitios web de documentación de frameworks o tecnologías referenciadas.
-

Enunciado:

Caso FIRSTstudent, Inc.

FIRSTstudent (<https://firststudentinc.com/>) es el mayor proveedor de servicios de transporte estudiantil en Norteamérica, con más de 900 millones de viajes estudiantiles al año.

Con sede en Cincinnati, Ohio, la empresa se dedica a brindar servicios de transporte seguros, confiables y rentables a distritos escolares, ciudades, empresas y sus ciudadanos. First Student es el líder indiscutible del mercado en transporte estudiantil y se centra en la prestación de servicios contratados de transporte del hogar a la escuela, incluyendo a estudiantes de educación especial, personas sin hogar y otras poblaciones estudiantiles.

Como proveedor líder de soluciones de transporte escolar en Norteamérica, First Student se esfuerza por brindar una atención inigualable y el viaje más seguro a la escuela a 5.5 millones de estudiantes cada día. Con un equipo de conductores altamente capacitados, la compañía completará mil millones de viajes estudiantiles durante el año escolar 2025. FIRSTstudent ofrece servicios confiables y de calidad, que incluyen transporte y gestión integral, transporte para necesidades especiales, electrificación de flotas, optimización de rutas y servicios de programación, mantenimiento y chárter con una flota de aproximadamente 45,000 autobuses.

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 FIRSTstudent.

El ecosistema de FIRSTstudent Manufacturing Platform requiere que el FIRSTstudent Web Application y el FIRSTstudent Mobile App cuenten con **Endpoints** en el RESTful API, para el manejo de la información de:

Buses, conformadas por los atributos *id*, *vehiclePlate*, *fuelTankType*, *districtId*, *totalSeats*.

Assignments, conformadas por los atributos *id*, *studentId*, *busId*, *assignedAt*.

Students, conformadas por los atributos *id*, *firstName*, *lastName*, *districtId*, *parentId*.

La tabla asociada a Students debe contener la siguiente información:

id	firstName	lastName	districtId	parentId
1	Emma	Smith	1	101
2	Liam	Smith	1	101
3	Olivia	Johnson	1	102
4	Noah	Johnson	1	102
5	Ava	Wilson	2	103
6	James	Wilson	2	103
7	Sophia	Anderson	2	104
8	William	Anderson	2	104
9	Isabella	Jackson	3	105
10	Lucas	Jackson	3	105

Como reglas de negocio, FIRSTstudent:

- No permite que se registre dos **buses** con el mismo **vehiclePlate**.
- Indica que los valores válidos para **districtId** son 1, 2 o 3.
- Indica que los valores válidos para **fuelTankType** son A, B, C o D.
- Indica que los valores válidos para **vehiclePlate** son tres caracteres alfabéticos en mayúsculas, un guión y cuatro dígitos (por ejemplo, ABC-1234).
- Especifica que los valores válidos para **totalSeats** son números enteros entre 20 y 40.
- No permite que se registre dos **assignments** con el mismo valor de **studentId**.
- Especifica que los **students** con el mismo **parentId** no deben asignarse a vehículos distintos.
- Indica que las asignaciones a un **vehicle** no deben exceder su **totalSeats**.
- Establece que la información que se desea preservar de cada **Bus**, incluye **id** (int, Primary Key, Autogenerado), **vehiclePlate** (string, Obligatorio, no vacío), **fuelTankType** (string, Obligatorio, no vacío), **districtId** (int, Obligatorio, no vacío), **totalSeats** (int, obligatorio, no vacío).
- Establece que la información que se desea preservar de cada **Student**, incluye **id** (int, Primary Key, Autogenerado), **firstName** (string, Obligatorio, no vacío), **lastName** (string, Obligatorio, no vacío), **districtId** (int, obligatorio), **parentId** (int, Obligatorio).
- Establece que la información de cada **Assignment**, incluye **id** (int, Primary Key, Autogenerado), **studentId** (int, Obligatorio), **busId** (int, Obligatorio), **assignedAt** (datetime, obligatorio).
- Especifica que un **Bus** puede ser referenciado en uno o muchos **Assignments**, pero un **Assignment** solo puede referenciar a un **Bus** en particular.
- Especifica que un **Assignment** puede referenciar a un **Student**, y un **Student** solo puede ser referenciado por un **Assignment** a la vez.

- Requiere que el valor de **assignedAt** sea poblado de forma automática al momento del registro con la fecha y hora actual.
- Indica que la información de Student se debe agregar de forma automática con los datos especificados previamente. Puede optar por implementar un método de tipo *findOrCreate* Student que se invoque durante el proceso de agregar un Assignment, o puede utilizar *UseSeeding* de Entity Framework Core (ver referencias).
- Especifica que tanto **Bus** como **Assignment** y **Student** pertenecen a diferentes bounded contexts, por lo que necesita un Anti-Corruption Layer (ACL) para que el bounded context que lo requiera, acceda a capacidades que exponga el otro bounded context como parte de la implementación de una característica requerida.
- Especifica que tanto **Bus** como **Assignment** son Aggregate Roots en sus respectivos bounded contexts, por lo que se requiere contar con atributos de auditoría para registrar **createdAt** (fecha y hora de creación de registro) y **updatedAt** (fecha y hora de última actualización de registro), de uso interno y poblados de forma automática por el sistema al momento de las operaciones de creación o actualización.

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

```
/api/v1/buses
/api/v1/buses/{busId}/assignments
```

Incluya como parte del desarrollo la implementación de todas las reglas de negocio, así como las especificaciones a nivel de requests y responses.

First Endpoint

Buses Endpoint (<https://localhost:{port}/api/v1/buses>)

Debe implementar **solo dos** operaciones en el RESTful API: agregar (POST) un **bus** y consultar (GET) un **bus** por **id**. Los valores de **id** son autogenerados al momento de almacenar la información. Al agregar se debe retornar en el response el status 201 (created). Para POST y para GET incluya en el response un objeto con el id generado para el bus y sus demás atributos. No incluya en el response los atributos de auditoría.

Second Endpoint

Assignments Endpoint (<https://localhost:{port}/api/v1/buses/{busId}/assignments>)

Debe implementar **solo una** operación sobre los **assignments** en el RESTful API: agregar (POST) un **assignment**. Al agregar, considere que busId es parte del path del request, por lo tanto, no debe ser parte del body del request. Se debe retornar en el response el status 201 (created) y el objeto incluyendo el id generado para el mismo y los demás atributos. No incluya en el response los atributos de auditoría.

Technical constraints

1. Elabore la solución con C#, .NET 9 y ASP.NET Core Framework.
2. Cree su solución y el proyecto de software con el nombre **eb<NRC>u< código-estudiante>.API** (por ejemplo, *eb4395u201621873.API*).
3. Considere que el aggregate **Bus** pertenece al bounded context **assets**, mientras que el aggregate **Assignment** y entity **Student** pertenecen al bounded context **operations**.
4. Considere el bounded context **shared** con los elementos que correspondan.
5. La información debe ser persistente en una base de datos relacional (MySQL), en un esquema **first_student**.
6. Aplique buenas prácticas de Arquitectura de Software, enfoque de Domain-Driven Design, separación en bounded contexts, layered architecture (Domain, Application, Interfaces, Infrastructure), patrones de strategic y tactical Domain-Driven Design, patrón CQRS, patrón Anti-Corruption Layer (ACL), principios y

patrones de diseño, convenciones de nomenclatura en **inglés**, así como buenas prácticas de nomenclatura en C# (entre ellas Upper Camel Case para Namespaces, Clases, Records, 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. Utilice minúsculas para los nombres de URL y términos compuestos separados por guión medio (-) para todos los endpoints.
8. Utilice Properties para representar los atributos de las clases Entity.
9. Utilice records en vez de clases para almacenamiento de valores inmutables.
10. Utilice el patrón Assembler para el Object Mapping en la sección Transform de interfaces layer.
11. Para **los** Aggregate Roots en cada bounded context, incluya properties de auditoría *CreatedAt* y *UpdatedAt* con valores generados de forma automática al momento de la creación. Utilice para ello la dependencia *EntityFrameworkCore.CreatedUpdatedDate*.
12. Utilice Fluent API para implementar las reglas de Object-Relational Mapping.
13. Documente su código en inglés, con **XML documentation comments**, colocando información de propósito para principales objetos de programación, así como propósito, parámetros y valor retornado en clases y métodos relevantes. Incluya como parte de la documentación sus nombres y apellidos como valor en un tag <remarks>.
14. Incluya documentación de Endpoints con OpenAPI.
15. Considere la gestión de excepciones en la aplicación.
16. Empaque su solución como un archivo **.zip. (único formato válido)** con el nombre **upc-pre-202402-si730-<sección>-eb-u<ódigo-estudiante>.zip** (por ejemplo, *upc-pre-202402-si730-wx59-eb-u201621873.zip*).
17. 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. First 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 con 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 <i>nombrada según especificaciones</i> en base de datos relacional según enunciado en el esquema <i>indicado</i> .	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. Second 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 con 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 <i>nombrada según especificaciones</i> en base de datos relacional según enunciado en el esquema <i>indicado</i> .	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. Cumple de forma completa con los technical constraints sobre organización.	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. Cumple con la mayoría de technical constraints sobre organización.	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. Cumple con solo algunos de los technical constraints sobre organización.	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, ó 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, ó 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. Cumple de forma completa con los technical constraints sobre nomenclatura.	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. Cumple con la mayoría de technical constraints sobre nomenclatura.	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. Cumple con solo algunos de los technical constraints sobre nomenclatura.	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, 14 de Julio del 2025

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>

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>

EntityFrameworkCore.CreatedUpdatedDate

<https://www.nuget.org/packages/EntityFrameworkCore.CreatedUpdatedDate>

Configuration options UseSeeding and UseAsyncSeeding methods

<https://learn.microsoft.com/en-us/ef/core/modeling/data-seeding>

What is a UUID (or GUID)?

<https://pddivine.medium.com/what-is-a-uuid-or-guid-16d0ead25008>

Guid struct

<https://learn.microsoft.com/en-us/dotnet/api/system.guid?view=net-8.0>