

<b>EVALUACIÓN</b>	Obligatorio	<b>GRUPO</b>		<b>TODOS</b>	28/10/2024
<b>MATERIA</b>	PROGRAMACIÓN 3				
<b>CARRERA</b>	Analista en Tecnologías de la Información / Analista Programador				
<b>CONDICIONES</b>	<p>- <b>Puntaje máximo:</b> 30 puntos</p> <p>- <b>Puntaje mínimo:</b> 1 punto</p> <p>- <b>Fecha de entrega:</b> 28/11/2024 hasta las 21:00 horas en <a href="http://gestion.ort.edu.uy">gestion.ort.edu.uy</a> (max. 40Mb en formato zip, rar o pdf)</p> <p><b>Uso de material de apoyo y/o consulta</b></p> <p><u>Inteligencia Artificial Generativa</u></p> <ul style="list-style-type: none"> <li>- Seguir las pautas de los docentes: Se deben seguir las instrucciones específicas de los docentes sobre cómo utilizar la IA en cada curso.</li> <li>- Citar correctamente las fuentes y usos de IA: Siempre que se utilice una herramienta de IA para generar contenido, se debe citar adecuadamente la fuente y la forma en que se utilizó.</li> <li>- Verificar el contenido generado por la IA: No todo el contenido generado por la IA es correcto o preciso. Es esencial que los estudiantes verifiquen la información antes de usarla.</li> <li>- Ser responsables con el uso de la IA: Conocer los riesgos y desafíos, como la creación de "alucinaciones", los peligros para la privacidad, las cuestiones de propiedad intelectual, los sesgos inherentes y la producción de contenido falso</li> <li>- En caso de existir dudas sobre la autoría, plagio o uso no atribuido de IAG, el docente tendrá la opción de convocar al equipo de obligatorio a una defensa específica e individual sobre el tema</li> </ul> <p><b>IMPORTANTE:</b></p> <ol style="list-style-type: none"> <li>1) Inscribirse</li> <li>2) Formar grupos de hasta 2 personas del mismo dictado</li> <li>3) Subir el trabajo a Gestión antes de la hora indicada (ver hoja al final del documento: "RECORDATORIO")</li> </ol> <p>Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con el Coordinador o Coordinación adjunta <b>antes de las 20:00hs.</b> del día de la entrega, a través de los mails <a href="mailto:alamon@ort.edu.uy">alamon@ort.edu.uy</a> y <a href="mailto:fernandez_ma@ort.edu.uy">fernandez_ma@ort.edu.uy</a>, o telefónicamente al 29021505 - int 1156 u 1138</p>				

Para la segunda parte del obligatorio deberemos implementar las siguientes **funcionalidades** exponiéndolas a través de los EndPoints apropiados de una Web API, y se consumirán desde un cliente MVC que use HttpClient, con el objetivo de testear su funcionamiento.

#### **RF1 – Login /Logout.**

Se implementará el Login y Logout. Se asegurarán los endpoints utilizando token JWT. Se sugiere implementar las funcionalidades pedidas sin control de acceso desde la API, y agregarlo posteriormente.

**Todas las funcionalidades que requieran autenticación serán realizadas por un usuario con rol Digitador.**

#### **RF2 – Listado de Atletas por Disciplina (API Web + HttpClient) – Sin autenticación**

- Crear un listado de atletas filtrado por disciplina. Este endpoint permitirá consultar todos los atletas que están registrados en una disciplina dado su Id. El listado se retornará ordenado alfabéticamente por nombre completo de atleta. Se incluirán como mínimo su Id (o número, según haya utilizado), su nombre completo y el nombre de su país.

#### **RF3– CRUD de disciplinas a través de Webapi. - Autenticación Jwt**

- Se podrán agregar disciplinas, modificarlas, eliminarlas, obtener una disciplina dado su Id, obtener una disciplina dado su nombre (recordar que es único), obtener todas las disciplinas.
- Estas operaciones se expondrán a través de una Web Api, y se probarán a través de un cliente HttpClient.

#### **RF4 – Control de cambios – Autenticación Jwt (heredada de la CRUD correspondiente).**

- Se deberá agregar control de cambios. Se debe registrar cualquier cambio, indicando la fecha en que se realizó, la operación (“Creado”, “Eliminado”, “Editado”), el tipo de entidad y el id de la entidad actualizada (dada de alta, dada de baja o modificada, no se registran las consulta) y el mail del usuario que realizó la operación.
- Por el momento solo se aplica a ABM de disciplina, pero debería poder aplicarse a cualquier entidad.
- Las entidades del dominio deberán ser agnósticas de que se las audita.

#### **RF5 – Búsqueda de Eventos por Disciplina, Fecha, Nombre del Evento y Rango de Puntaje (API Web + HttpClient) – Autenticación Jwt**

- Obtener un listado de eventos utilizando múltiples criterios de búsqueda. El usuario podrá filtrar los eventos por disciplina, por fecha de comienzo, por el nombre del evento y por un rango de puntaje obtenido por los atletas participantes.
  - o El sistema permitirá filtrar eventos por:
    - Disciplina: seleccionada por su ID.
    - Rango de fechas: se podrán ingresar una fecha de inicio y una fecha de fin. Si ambas fechas son iguales, se buscarán los eventos que se realicen ese día.
    - Nombre del evento: se podrá realizar una búsqueda parcial por el nombre de la prueba (por ejemplo, "salto").

- Rango de puntaje: se podrá especificar un puntaje mínimo y máximo, y se mostrarán los eventos donde al menos un atleta haya obtenido un puntaje en ese rango (incluyendo topes).

### Se pide:

1. Implementar lo necesario para exponer los endpoints de una Web Api Restful correspondiente a las funcionalidades anteriores.
2. Incluir la documentación de la Webapi (deberá poder verse desde Swagger).
3. Resolver los requisitos anteriores mediante el uso de HttpClient desde una aplicación MVC que consuma los endpoints de la WebApi.
2. Realizar precarga de datos mediante la ayuda de Chat GPT. Se deberán incluir en la documentación la captura y el link a la secuencia de los prompts utilizados. Se requieren como mínimo 30 registros de disciplinas, 100 atletas, 80 eventos. Los datos deberán ser fácilmente identificables y coherentes con la realidad planteada, y aptos para resolver todas las búsquedas solicitadas.  
Esos datos deberán figurar en el script de los insert que se entrega.

### ENTREGA FINAL EN GESTIÓN:

Se subirá a gestión un único archivo comprimido que contenga:

#### 1. IMPLEMENTACIÓN.

Archivo comprimido en formato .zip, .rar o .7z con la solución funcionando completa en .NET 8.0, usando Entity Framework Core 8.0 y servicio WebApi. El MVC deberá ser una solución independiente, y solo podrá acceder a las funcionalidades mediante HttpClient. **El cliente MVC no tiene acceso directamente a la BD, y al ser solamente un prototipo para testear la API no debe implementar arquitectura limpia ni DDD.**

#### 2. Precarga de datos.

Se precargarán datos en la base según se mencionó anteriormente. Si requiere datos que no estén explícitamente indicados deberá incluirlos, también.

#### 3. PDF con la documentación siguiente:

- a. Carátula con el nombre, número de estudiante y foto de los integrantes del grupo.
- b. Tabla de contenido con la página en que se ubica cada ítem de la documentación.
- c. Diagrama de casos de uso
- d. Diagrama de clases (UML) para todas las capas implementadas. Se excluyen las Views de MVC. Los diagramas deberán ser legibles. Se deberá incluir la Web Api.
- e. Código fuente de toda la aplicación (ambas soluciones, MVC y Webapi con librerías de clase) incluido en la documentación.

- f. Archivo Astah con los diagramas anteriormente solicitados.
- g. Scripts con los juegos de datos de prueba.

Cuando no se haya explicitado la cantidad de registros previamente en el presente documento, se espera que el juego de datos tenga al menos 10 registros de cada tabla. Se incluirán tanto los esquemas como los insert con los datos. Se deberán incluir tablas y filas tanto para aquellas entidades que no deben registrarse a través del sistema (las precargas explicitadas anteriormente) como para aquellos registros que surgen del sistema en funcionamiento.).

### Reglas generales:

- Las magnitudes serán positivas o cero.
- A menos que se especifique lo contrario todos los datos son requeridos.
- Cuando corresponda se deberán implementar Value Objects.
- Las validaciones, errores, etc. se manejarán a través de excepciones.
- Todas las entidades se identificarán mediante un Id autonumérico secuencial (otorgado por la BD, inmutable).
- En la aplicación Web cada vez que se produzca un error en las ABM, o cuando no se obtengan resultados para las solicitudes de información se deberá informar convenientemente al usuario. **En el caso de las Web Api se deberán retornar los status code adecuados según la guía REST.**
- Cuando sea necesaria información para la operativa y su registro no sea explícitamente solicitado en los requerimientos, será cargada manualmente en la base de datos.
- Las consultas se realizarán a través de LINQ **exclusivamente**.
- Se deberán utilizar DTO.
  
- **En el caso de utilizar código generado mediante inteligencia artificial generativa, deberá estar en condiciones de explicarlo en la defensa y deberá incluir el link a los prompts utilizados.**

Este proyecto se realizará usando biblioteca de clase, proyecto Web MVC, proyecto Web Api de .NET 8, C# como lenguaje de programación, Entity Framework 8 para el acceso a la base de datos y Linq como lenguaje de consulta. Se preferirá sintaxis de método.

Se deberá organizar el backend siguiendo las recomendaciones de Arquitectura Limpia (Clean Architecture) y DDD (Domain Driven Design).


---

## RECORDATORIO: IMPORTANTE PARA LA ENTREGA

- **Obligatorios**

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de la misma.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. Ingresá al sistema de Gestión.
2. En el menú, seleccioná el ítem "Evaluaciones" y la instancia de evaluación correspondiente, que figura bajo el título "Inscripto".
3. Para iniciar la entrega hacé clic en el ícono: 
4. Ingresá el número de estudiante de cada uno de los integrantes y hacé clic en "Agregar". El sistema confirmará que los integrantes estén inscriptos al obligatorio y, de ser así, mostrará el nombre y la fotografía de cada uno de ellos. Una vez agregados todos los integrantes, hacé clic en "Crear equipo".

**Cualquier integrante podrá:**

- **Modificar la integración del equipo.**
- **Subir el archivo de la entrega.**

5. Seleccioná el archivo que deseás entregar. Verificá el nombre del archivo que aparecerá en la pantalla y hacé clic en "Subir" para iniciar la entrega. Cada equipo (hasta 2 estudiantes) debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar). El archivo a subir debe tener **un tamaño máximo de 40mb**

Cuando el archivo quede subido, se mostrará el nombre generado por el sistema (1), el tamaño y la fecha en que fue subido.

6. El sistema enviará un e-mail a todos los integrantes del equipo informando los detalles del archivo entregado y confirmando que la entrega fue realizada correctamente.
7. Podés cerrar la pestaña de entrega y continuar utilizando Gestión o salir del sistema.
8. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
9. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc).
10. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con la Coordinadora o Coordinación adjunta antes de las 20:00hs. del día de la entrega, a través de los mails, [alamon@ort.edu.uy](mailto:alamon@ort.edu.uy), [martinez\\_i@ort.edu.uy](mailto:martinez_i@ort.edu.uy) o [fernandez\\_ma@ort.edu.uy](mailto:fernandez_ma@ort.edu.uy); o telefónicamente al 29021505 - int 1156 u 1138