

<b>EVALUACIÓN</b>	Obligatorio 1	<b>GRUPO</b>	Todos	<b>FECHAS</b>	Lectura: Semana 3 Entrega: Semana 9
<b>MATERIA</b>	Programación 3				
<b>CARRERA</b>	Analista Programador – Analista en Tecnologías de la Información				
<b>CONDICIONES</b>	<p>- Puntos: Máximo: 30 Mínimo: 0 - <b>Fecha máxima de entrega: 13/10/2022</b></p> <p><b>LA ENTREGA SE REALIZA EN FORMA ONLINE EN ARCHIVO NO MAYOR A 40 MB EN FORMATO ZIP, RAR O 7zip. EXCLUSIVAMENTE A TRAVÉS DEL SITIO DE GESTIÓN (<a href="http://gestion.ort.edu.uy">gestion.ort.edu.uy</a>)</b></p> <p><b>IMPORTANTE:</b></p> <ul style="list-style-type: none"><li>- Inscribirse</li><li>- Formar grupos de hasta tres personas.</li><li>- Subir el trabajo a Gestión antes de la hora indicada, ver hoja al final del documento: "RECORDATORIO"</li></ul>				

#### INFORMACIÓN IMPORTANTE:

- Las consultas sobre la letra deberán realizarse a través del foro de aulas.
- La no asistencia a la defensa implica la pérdida de puntos del obligatorio para el estudiante. En caso de fuerza mayor deberá avisar al docente mediante mail y a coordinación por Teams para analizar si se autoriza la programación de una nueva fecha.
- Si el tamaño del archivo comprimido que entrega supera los 40Mb puede eliminar de la solución las carpetas bin, y obj preservando los archivos propios que pueda haber incluido en estas carpetas (ej.: archivos de texto, imágenes, etc.). Se recomienda enfáticamente mantener una copia de la versión completa para prevenir problemas durante la descarga de paquetes.
- **El formato de la defensa se definirá previo a la entrega.**
- **Las aclaraciones realizadas por los docentes a través del foro de aulas serán consideradas parte integral de la letra, por lo que es responsabilidad de los estudiantes acceder asiduamente al foro.**

En el marco del próximo mundial de Qatar, nos han contratado para desarrollar un sistema de apuestas en línea sobre los resultados de los partidos jugados. Por el momento debemos presentar en un prototipo que incluya:

Nota: Tomar en cuenta que a todas las entidades se les asignará identificador autonumérico secuencial al registrarlas.

1. **Operaciones CRUD de países participantes (Repositorios/Aplicación de consola o web a elección).**

Para cada **país** se registrará su **nombre**, su **código ISO-Alfa3**, **PBI per cápita**, su **población**, una **imagen con su bandera**, y la **región**. Las regiones son las correspondientes a los 5 continentes.

De cada **región** se guarda el **nombre** del continente, y tanto el nombre y código ISO-Alfa3 del país son únicos.

El nombre del país sólo podrá contener caracteres alfabéticos y espacio embebido, el código alfa3 debe tener exactamente 3 caracteres y debe comenzar con la primera letra del país. El nombre de la región debe ser uno de la siguiente lista: "África", "América", "Asia", "Europa" y "Oceanía".

Todos los números deben ser positivos.

Las imágenes de las banderas deberán precargarse en una carpeta del servidor, y tendrán como nombre el código ISO-alfa3 del país correspondiente y extensión .png (el formato también debe ser png).

La información será almacenada en una base de datos Sql Server utilizando Entity Framework, y cada vez que sea utilizada (por ejemplo, para dar de alta las selecciones o visualizarlas) se cargarán todos los países a memoria.

Se deberá contar al menos con información de 12 países.

Además de permitir el **alta de país**, será posible **modificar sus datos**, **eliminar un país**, **buscar un país por su Id**, **buscar un país por su código alfa-3**, **obtener todos los países**, **obtener los países de una región**. Para eliminar un país se deberá verificar que no tenga registros relacionados.

**Nota:** Provisoriamente, para realizar las pruebas mientras no se haya visto el tema Entity Framework, se pueden precargar en el repositorio correspondiente los países.

Una vez que se esté utilizando Entity Framework, se materializarán todos los países y serán manipulados desde las listas y no desde Entity Framework cuando se requiera hacer consultas.

## 2. CRUD de selección clasificada (Web Api – Testeada desde Postman o equivalente).

Como es conocido, la federación de cada uno de los países participantes presenta su selección. A cada **selección** participante se le asignará su **país**, el **nombre**, **email** y **teléfono** de la persona de contacto, y la **cantidad de potenciales apostadores**. La cantidad de potenciales apostadores no está relacionada con la población del país. La selección pertenece a un **grupo**, que se reconoce por una **letra ("A" - "H")**.

Se deberá verificar que haya una sola **selección** por país, que el nombre de la persona de contacto tenga solamente caracteres alfabéticos y espacios embebidos, que el email sea válido según las reglas habituales, y el teléfono tenga por lo menos 7 caracteres numéricos.

Todas las cantidades numéricas deberán ser positivas.

También se podrá dar de **baja una selección**, **modificar sus datos**, **buscar una selección dado su Id** y **obtener la lista de todas las selecciones**.

Las selecciones no se podrán dar de baja si tienen registros relacionados.

Estas funcionalidades se realizarán utilizando Web Api y Entity Framework. La Api se consumirá desde Postman.

### 3. Registro del fixture de la fase de Grupos. (Web Api – Testeada desde Postman o equivalente)

El mundial se juega en dos fases, fase de grupo y **fase eliminatoria**.

En esta etapa solamente se desarrollará un prototipo para la **fase de grupos**. **No obstante, al modelar la solución se deberá considerar que en un futuro será necesario incluir la fase de eliminación, que tiene atributos y reglas diferentes.**

En esta fase juegan todos contra todos dentro de su grupo (c/u 3 partidos). Para eso se definieron 8 grupos formados por 4 selecciones cada uno. En cada grupo juegan todos contra todos en las 3 jornadas (fechas) del grupo.

Como es sabido, en cada **partido** juegan 2 **selecciones**, en una **fecha y hora** determinadas.

Aunque los partidos ya fueron definidos en el sorteo realizado por la FIFA, a los efectos de este prototipo no será necesario registrar los partidos reales.

Al ingresar la información de un partido se deberá controlar que cada selección en esta etapa juegue solo 3 partidos, siempre con un contrincante diferente, que el partido quede entre las dos fechas tope de la fase de grupos fijadas por defecto entre el 20 de noviembre 2022 y 2 de diciembre 2022.

Cada día se disputarán hasta un máximo de cuatro partidos, en cualquiera de los cuatro horarios diferentes previstos, lo que permitirán ver todos los partidos. Los horarios serán a las 07:00, 10:00, 13:00 y 16:00 de Uruguay. Estos horarios son fijos, y por razones de televisión no se pueden cambiar. Tampoco podrán superponerse dos partidos en el mismo horario.

Al registrar los equipos que juegan cada partido se deberá verificar que un equipo no juegue contra sí mismo y que el contrincante pertenezcan al mismo grupo. Para este prototipo no se tomarán en cuenta otras restricciones que pudieran formar parte del reglamento de la FIFA.

Se deberá implementar la Web api que permita el **registro del partido**, y se consumirá desde Postman.

### 4. Registrar **resultados** e incidencias de los partidos. (Web Api – Testeada desde Postman o equivalente)

Cuando finalice un partido se indicará la cantidad de **goles** de cada equipo.

En la fase de grupos puede haber empate. También interesará conocer la cantidad de tarjetas **rojas (por acumulación de 2 amarillas)**, de **rojas directas** y **amarillas** obtenidas por cada selección.

A cada equipo se **le asignan puntos** según un sistema de ligüilla, donde la victoria valdrá tres puntos, el empate un punto, y la derrota, cero. Con esta información será posible conocer el **puntaje** final de cada partido.

En este prototipo para realizar los controles necesarios para registrar el puntaje, se cargarán las listas con todas las selecciones y con los partidos, no se realizarán consultas directamente en la base de datos a través de Linq.

Cuando se verifiquen las condiciones necesarias, se almacenará el resultado/incidencia en la base de datos. Todos los numéricos son cero o positivos.

#### **5. Obtener la información de un grupo con su fixture. (Web Api – Presentación Web utilizando MVC)**

Se solicita el nombre de grupo y se **despliegan los datos de los partidos de ese grupo** con todos los datos de sus fechas, selecciones (todos sus datos), y si los partidos ya se jugaron se incluirán sus resultados e incidencias (cantidad de goles a favor, cantidad de goles en contra, cantidad de amarillas, cantidad de rojas por acumulación de 2 amarillas, cantidad de rojas directas).

#### **PRE-ENTREGA:**

**En una fecha indicada por el docente - plazo máximo 2 semanas a partir de la lectura - deberá subir a una tarea de aulas el diagrama que modele la solución de dominio y persistencia. Deberá incluir las clases con sus atributos, relaciones, y métodos con firmas completas. Este documento se entrega a los únicos efectos de constatar el avance del proyecto.**

**El diagrama que se solicita para la entrega final en Gestión puede incluir las modificaciones que considere necesarias.**

**Si bien no habrá una devolución sobre esta entrega, podrá realizar consultas sobre el modelado de su sistema exclusivamente hasta la fecha máxima en que pueda subirlo a aulas.**

#### **ENTREGA:**

#### **DOCUMENTACIÓN:**

Se incluirán en el archivo comprimido que se entrega en Gestión un único **PDF** conteniendo:

- Carátula con el nombre, número de estudiante y foto de los integrantes del grupo.
- Tabla de contenido con la página en que se ubica cada ítem de la documentación.
- Los documentos y diagramas solicitados debajo (legibles).
- a) Diagrama de los casos de uso detectados para esta etapa. Aunque no se pide para esta etapa la presentación de la mayor parte de los requerimientos, se considerará como si cada requerimiento fuera accedido desde la web. Todo lo que implique modificaciones al sistema podrá ser realizado por un usuario con rol admin – que podrá acceder a todas las funcionalidades, incluso listados. El fixture también podrá ser accedido por cualquiera que navegue al sitio, y al ranking tendrán acceso además del admin, los usuarios “apostadores”.
- b) Diagrama de clases de todas las capas que incluya en su solución.  
Se incluirán:

- Clases de persistencia / Repositorios / api / capa de aplicación
- Clases del dominio, incluyendo todas las entidades de negocio que modelen la realidad planteada.
- Se deberán indicar los atributos, relaciones, adornos relevantes de las relaciones, y **los métodos que definan su comportamiento según una correcta asignación de responsabilidades.**
- Se deberá aplicar una correcta división en capas según Clean Architecture
- Se incluirá también un único archivo ASTAH con los diagramas.

**IMPORTANTE:** No se considerarán documentaciones que no estén entregadas en los formatos planteados (PDF y Astah), que no contengan la carátula con la identificación de los integrantes del equipo y/o que no estén incluidas dentro del único archivo comprimido entregado en gestión.

#### **IMPLEMENTACIÓN:**

- a) Todas las funcionalidades descritas anteriormente (1 al 5).**

**Solución implementada en Visual Studio 2019 utilizando .NET Core 3.1, que incluya el proyecto web api, y las bibliotecas de clase necesarias, utilizando como lenguaje de programación C# siguiendo el esquema de capas propuestas según Clean Architecture visto en el curso.** Se deberán incluir las validaciones que correspondan durante los ingresos para garantizar la consistencia del sistema, estén o no explicitadas en la letra. Se recomienda incluir comentarios al código para comprensión de la lógica más importante. La base de datos será SQL Server y el proveedor utilizado deberá ser SqlClient y se persistirá a través de Entity Framework.

- b) Datos de prueba:**

Se entregarán los scripts con la creación de las tablas y los insert suficientes para probar la aplicación durante la defensa y corrección. Deberá haber al menos diez instancias de cada entidad.

Tomar en cuenta que en algunos casos las diez instancias de casos de prueba pueden no ser suficientes, por lo que deberá incluir más.

**La no presentación de los datos de prueba, o su insuficiencia para probar la aplicación en la defensa serán valorados negativamente.**

### RECORDATORIO: IMPORTANTE PARA LA ENTREGA

➤ **Obligatorios** (Cap.IV.1, Doc. 220)

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. La entrega se realizará desde [gestion.ort.edu.uy](http://gestion.ort.edu.uy)
2. Previo a la conformación de grupos cada estudiante deberá estar inscripto a la evaluación. **Sugerimos realizarlo con anticipación.**
3. **Cualquier integrante del grupo de obligatorio podrá subir** la entrega
4. Cada equipo (3 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)
5. El archivo a subir debe tener **un tamaño máximo de 40mb**
6. Les sugerimos **realicen una 'prueba de subida' al menos un día antes**, donde conformarán el **'grupo de obligatorio'**.
7. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
8. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc.)
9. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor pasar por la oficina del Coordinador o por Coordinación adjunta **antes de las 20:00hs.** del día de la entrega

Si tuvieras una situación particular de fuerza mayor, debes dirigirte con suficiente antelación al plazo de entrega, al Coordinador de Cursos o al Secretario Docente.