**EVALUACIÓN CORRESPONDIENTE AL PRIMER PARCIAL**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FACULTAD: | **Tecnología Informática** | | | | |
| CARRERA: | **Analista programador** | | | | |
| ALUMNO/A: | **Víctor Alberto Terrile** | | | | |
| SEDE: | **UAI** | | LOCALIZACIÓN: | **Remoto** | |
| ASIGNATURA: | **Lenguajes de última generación** | | | | |
| CURSO: | **2º B-D** | | TURNO: | **Noche** | |
| PROFESOR: | **Arigós Gonzalo** | | FECHA: | **29/09/2024** | |
| TIEMPO DE RESOLUCIÓN: | | **180’** | EXAMEN PARCIAL NRO: | | **1** |
| MODALIDAD DE RESOLUCIÓN: **A Distancia / Individual** | | | Presencial / Virtual / Escrito / Oral / Individual / Grupal | | |
|  | | |  | | |
| RESULTADOS DE APRENDIZAJE: | | | | | |

**Propósito:**

Evaluar la capacidad del estudiante para diseñar y desarrollar un programa orientado a objetos. Su habilidad para administrar el tiempo y los recursos seleccionados para el logro del objetivo propuesto, su capacidad para integrar el marco teórico propuesto con los resultados alcanzados.

CRITERIOS DE EVALUACIÓN:

* Creatividad y originalidad de la propuesta.
* Claridad en la organización de la escritura del código y los comentarios incluidos en el mismo.
* Precisión en el resultado obtenido.
* Utilización de los conceptos abordados en clase: Definición de clase abstractas, selladas y concretas. Propiedades, métodos, eventos, constructores y destructores. Herencia, polimorfismo, agregación y asociación. Sobreescritura y sobrecarga.
* Racionalidad y coherencia en la validación de los datos ingresados y obtenidos, así como en el control de las excepciones

El examen se considerará aprobado con una nota de 4 (cuatro) que se obtendrá con el 60% de los ítems del PROYECTO correctamente desarrollados.

**Guía de Resolución:**

**Título de proyecto software:**

PartidosGestión: Aplicación para gestionar eventos deportivos

**1. Introducción**

Nos solicitan desarrollar una app para gestionar la carga de partidos de futbol, básquet y tenis.

El cliente necesita la posibilidad de dar de alta nuevos partidos y luego poder modificar el marcador de goles/ puntos de cada partido que fue cargado.

**2. Marco de Referencia**

La solicitud es realizada por una casa de apuestas online que necesita mantener actualizada su base de datos de partidos para determinar los resultados obtenidos por sus usuarios.

**3. Objetivos**

Los objetivos del programa son:

1. Poder crear un partido y persistirlo en una base de datos.
2. Validar los datos ingresados por el usuario.
3. Permitir que el usuario visualizar todos los partidos cargados.
4. Permitir la eliminación de un partido existente.
5. Permitir modificar el marcador de goles/puntos de los partidos cargados.

**4. Relevamiento de Datos**

Se solicita registrar:

Partido:

* ID\_PARTIDO -> PK Auto incremental
* ID\_DEPORTE -> FK de tabla Deporte
* EQUIPO\_LOCAL -> VARCHAR
* EQUIPO\_VISITANTE -> VARCHAR
* FECHA\_REGISTRO -> DATETIME
* FECHA\_PARTIDO -> DATETIME
* MARCADOR\_LOCAL -> INT
* MARCADOR\_VISITANTE -> INT

Deporte:

* ID\_DEPORTE -> PK Auto incremental
* DESCRIPCION -> VARCHAR

Se solicita precargar los siguientes deportes en la tabla:

* Futbol
* Básquet
* Tenis

Estos datos deben ser cargados en una tabla Deporte y obtenidos al momento de la carga del formulario.

Crear entidades de negocio “Entrada” y “Deporte” en la solución de código a entregar.

El sistema tendrá la capacidad de validar los siguientes datos al momento del alta de un nuevo partido:

Equipo local y equipo visitante

* No puede estar vacío.
* La cantidad de caracteres debe ser mayor a 5.

Fecha del partido:

* La fecha no puede ser menor a la fecha actual.

Deporte:

* No debe estar vacío

Marcador local y visitante:

* Al dar de alta un partido arrancan por defecto en cero (luego podrán ser modificados)
* El valor no puede ser menor a cero.

Fecha de registro:

* Utilizar la fecha actual del sistema

Para todos los casos, se pide:

* Desarrollar las bases de datos y los formularios necesarios
* Utilizar adecuadamente herencia, polimorfismo y sobrecarga de métodos.
* Debe ser una aplicación en capas. Las clases deben estar en proyectos separados
* Capturar todos los errores.
* Utilizar SQLParameters.
* Utilizar siempre objetos de negocio en las capas de Presentación / BLL (no dataset)
* Utilizar todas las buenas prácticas vistas en clase

**5. Plan de trabajo**

Prever una interfaz de usuario (Windows Form) que permita instanciar la clase "Partido".

¿Qué datos serán provistos por el usuario?

* Deporte
* Equipo local
* Equipo visitante
* Fecha de partido

¿Qué datos serán administrados por el sistema?

* ID PARTIDO -> ID Auto incremental
* Fecha de registro -> fecha de hoy del sistema

¿Qué acciones podrá hacer el usuario?

* Dar de alta un partido indicando: deporte, equipo local y visitante y fecha de partido
* Consultar y visualizar todos los partidos dados de alta
* Eliminar un partido a través del ID del a mismo (Actualizar grilla luego de eliminar)
* Modificar el marcador local y visitante de un partido a partir del ID del mismo (Actualizar grilla luego de modificar)   
  **Solo se puede modificar el marcador de partidos que se jueguen el día de hoy, por lo tanto, se solicita: Para poder modificar el marcador de un partido, la fecha del partido debe ser igual a la fecha de hoy, de lo contrario arrojar excepción informando lo sucedido**

**Condiciones mínimas de aprobación:**

* Utilizar arquitectura en capas (UI, BLL, DAL, Entity).
* Aplicar buenas prácticas de la programación orientada a objetos.
* Aplicar buenas prácticas en las capas de negocio y acceso a datos.
* Las cadenas de conexión deben estar embebidas en archivos de configuración.
* Utilizar excepciones en caso de que no se cumpla una regla de negocio y capturar todos los errores en bloques try/catch.
* Liberar correctamente los recursos mediante el uso de bloques using.
* Utilizar SqlParameters para queries con datos de negocio.
* Desarrollar la base de datos en SQL Server.
* El programa se debe entregar sin errores de compilación, comprimido en formato zip o rar, y debe incluir los scripts de instalación de la base de datos, incluyendo los datos de prueba utilizados durante el proceso de desarrollo de la evaluación.

**Formulario de ejemplo:**

