



# 🎵 Que vengan los bomberos 🎵

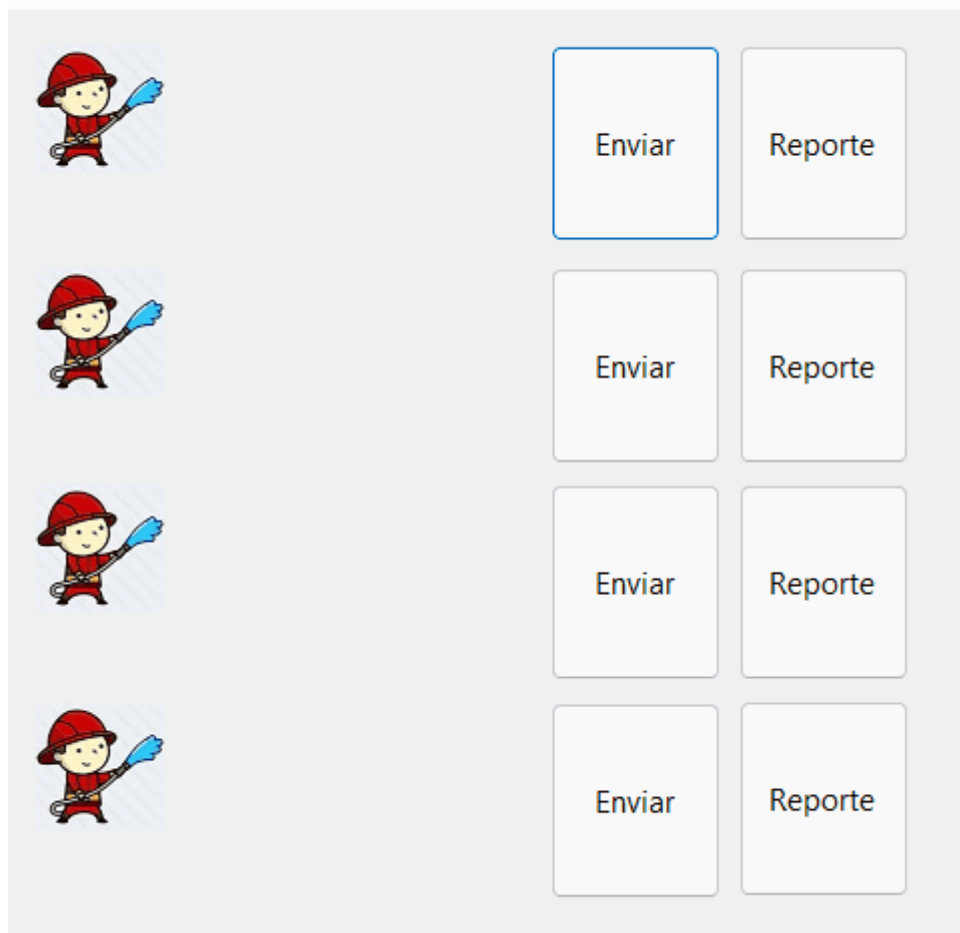
Instancia de evaluación	Cuatrimestre	Profesor	Tiempo de resolución
Segundo Parcial	2do 2020	F. Dávila	120 minutos

## Criterios de evaluación

- **2 (dos) errores en el mismo tema anulan su puntaje.**
- La correcta documentación y reglas de estilo de la cátedra serán evaluadas.
- Colocar sus datos personales en el nombre de la carpeta principal y la solución: *Apellido.Nombre.Div. Ej: Pérez.Juan.2D*. No se corregirán proyectos que no sea identificable su autor.
- No se corregirán exámenes que no compilen.
- **Reutilizar** tanto código como crean necesario.
- Colocar nombre de la clase (en estáticos), `this` o `base` en todos los casos que corresponda.
- Aplicar los principios de los 4 pilares de la programación orientada a objetos.

## Consigna

Se realizará una aplicación para despachar bomberos a distintos servicios.



1. Empezar a trabajar desde [la cáscara](#).

2. Crear la base de datos `bomberos-db` y ejecutar en ella el siguiente script:

```
USE [bomberos-db]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[log](
[id] [int] IDENTITY(1,1) NOT NULL,
[entrada] [varchar](100) NOT NULL,
[alumno] [varchar](60) NOT NULL
) ON [PRIMARY]
GO
```

3. Cambiar el nombre de la carpeta de la solución con sus datos personales: *Apellido.Nombre*.

4. Crear un proyecto de biblioteca de clases llamado `Entidades` y agregar los elementos del siguiente diagrama:

**Bombero**  
Class

Fields

- nombre : string
- salidas : List<Salida>

Methods

- AtenderSalida(int bomberoIndex) : void
- Bombero(string nombre)
- Guardar(Bombero info) : void
- IArchivo<string>.Guardar(string info) : void
- IArchivo<string>.Leer() : string
- Leer() : Bombero

Events

- MarcarFin : FinDeSalida

**Salida**  
Class

Fields

- fechaFin : DateTime
- fechaInicio : DateTime

Properties

- FechaFin { get; set; } : DateTime
- FechaInicio { get; set; } : DateTime
- TiempoTotal { get; } : double

Methods

- FinalizarSalida() : void
- Salida()

**IArchivo<X>**  
Generic Interface

Methods

- Guardar(X info) : void
- Leer() : X

**FinDeSalida**  
Delegate

bomberoIndex : int

5. Extender la clase `DateTime` para calcular la diferencia en segundos entre una fecha de inicio y una fecha de fin:

```
(fin - inicio).TotalSeconds;
```

6. Clase `Salida`:

- El constructor dará inicio a la salida colocando la fecha y hora actual en el atributo correspondiente.
- El método `FinalizarSalida` dará fin a la salida colocar la fecha y hora actual en el atributo correspondiente.
- La propiedad `TiempoTotal` retornará la diferencia en segundos entre la fecha de inicio y la fecha de fin.

7. Crear un nuevo tipo de excepción llamado `BomberoOcupadoException` con al menos dos sobrecargas de constructores.

8. Clase `Bombero`:

- i. Implementará la interfaz `IArchivos` tantas veces como muestra el diagrama. Esto requerirá implementar la interfaz de forma implícita y de forma explícita.
  - a. Para `Bombero`: serializará y deserializará el objeto en formato XML. Almacenarlo en un archivo `Bombero.xml` en el escritorio.
  - b. Para `string`: guardará y leerá de la base de datos el mensaje recibido como parámetro. Al leer concatenar todos las entradas del log en un solo `string`.
- ii. El método `AtenderSalida` realizará lo siguiente:
  - a. Agregará una nueva salida a la lista del bombero.
  - b. Suspenderá el hilo entre 2 y 4 segundos.
  - c. Finalizará la salida.
  - d. Registrará la salida (horario de inicio, horario de fin y duración total) en la tabla `log` de la base de datos.
  - e. Avisará mediante el evento `MarcarFin` que se terminó la salida. Utilizar el parámetro `bomberoIndex` para informar al formulario cuál bombero fue.

9. Formulario `Cuartel`:

- i. El método `DespacharServicio` deberá ejecutar `AtenderSalida` en un nuevo hilo.
- ii. El manejador del evento `FormClosing` del formulario deberá asegurarse que no quede ningún hilo activo al salir.
- iii. El método `OcultarBombero` será el manejador del evento `MarcarFin`. Hacer todo lo necesario para que funcione.
- iv. Si se quiere despachar a un bombero que ya está atendiendo otra salida, se deberá lanzar la excepción `BomberoOcupadoException` y controlarla en el evento `Click` de cada botón, guardando en el log "*Salida bombero X no concretada*".
- v. Los botones `Reporte` serializarán el objeto `Bombero` correspondiente.

10. Pruebas unitarias:

- i. Verificar que los métodos `Leer` y `Guardar` de `Bombero` serialicen y deserialicen correctamente.
- ii. Verificar que el método de extensión de `DateTime` funcione tal como se espera.

## Resolución



Video



Código

Last updated on **11/24/2021** by ***mauriciocerizza***