



Universidad Tecnológica Nacional
Facultad Regional Avellaneda

Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	04/08/2022
Nombre:		Docente ⁽²⁾ :	
División:	2º	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	PP	RPP	SP
		RSP	FIN
			X

(1) Las instancias validas son: 1º Parcial (**PP**), Recuperatorio 1º Parcial (**RPP**), 2º Parcial (**SP**), Recuperatorio 2º Parcial (**RSP**), Final (**FIN**). Marque con una cruz.

(2) Campos a ser completados por el docente.

IMPORTANTE:

- **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 del proyecto principal, colocando: Apellido.Nombre.Departamento. Ej: Pérez.Juan.2D. No se corregirán proyectos que no sea identificable su autor.
- **TODAS** las clases deberán ir en una Biblioteca de Clases llamada Entidades.
- 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.
- Colocar el nombre del docente con el cual cursó.

TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 90 MINUTOS.

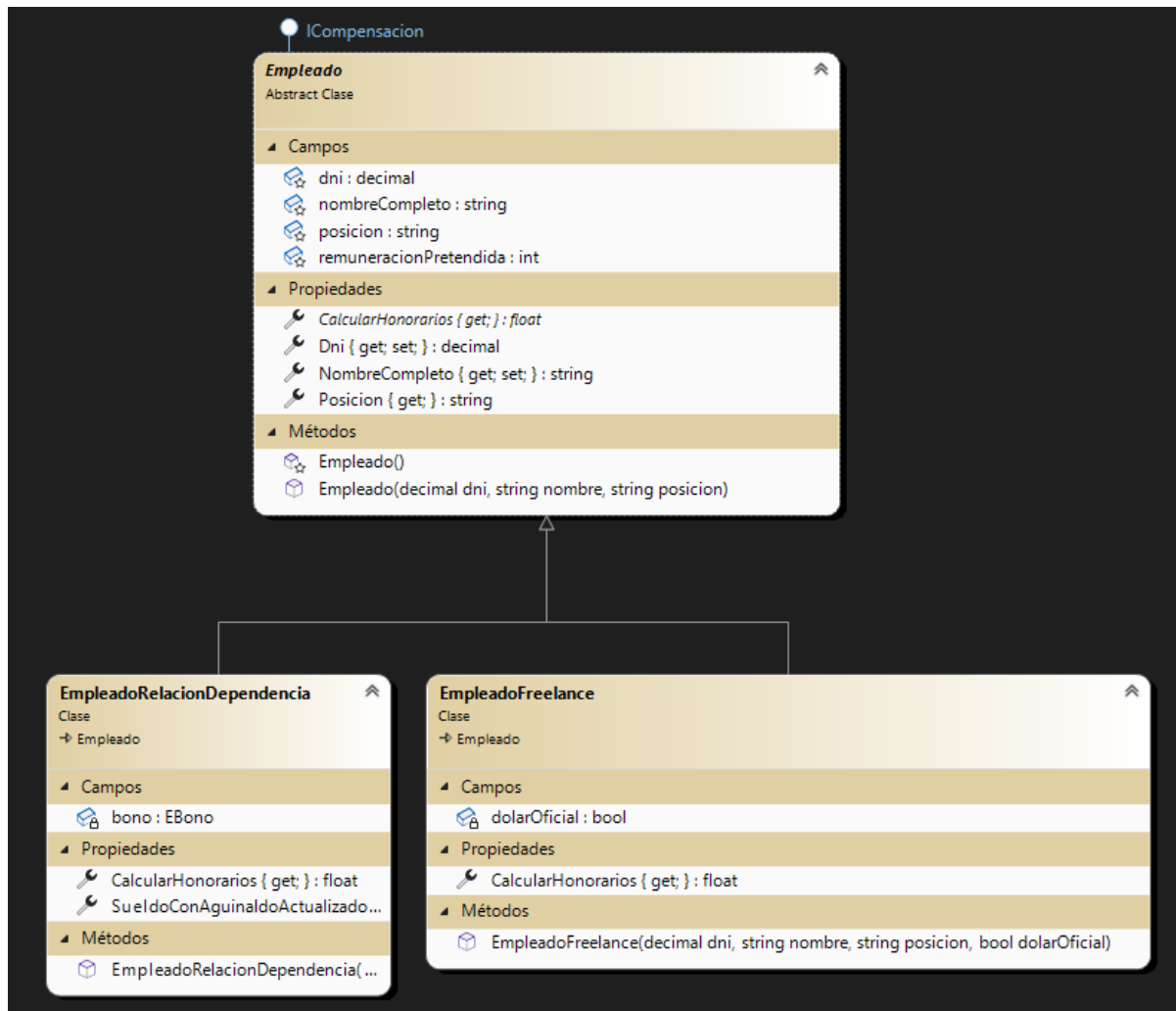
Introducción

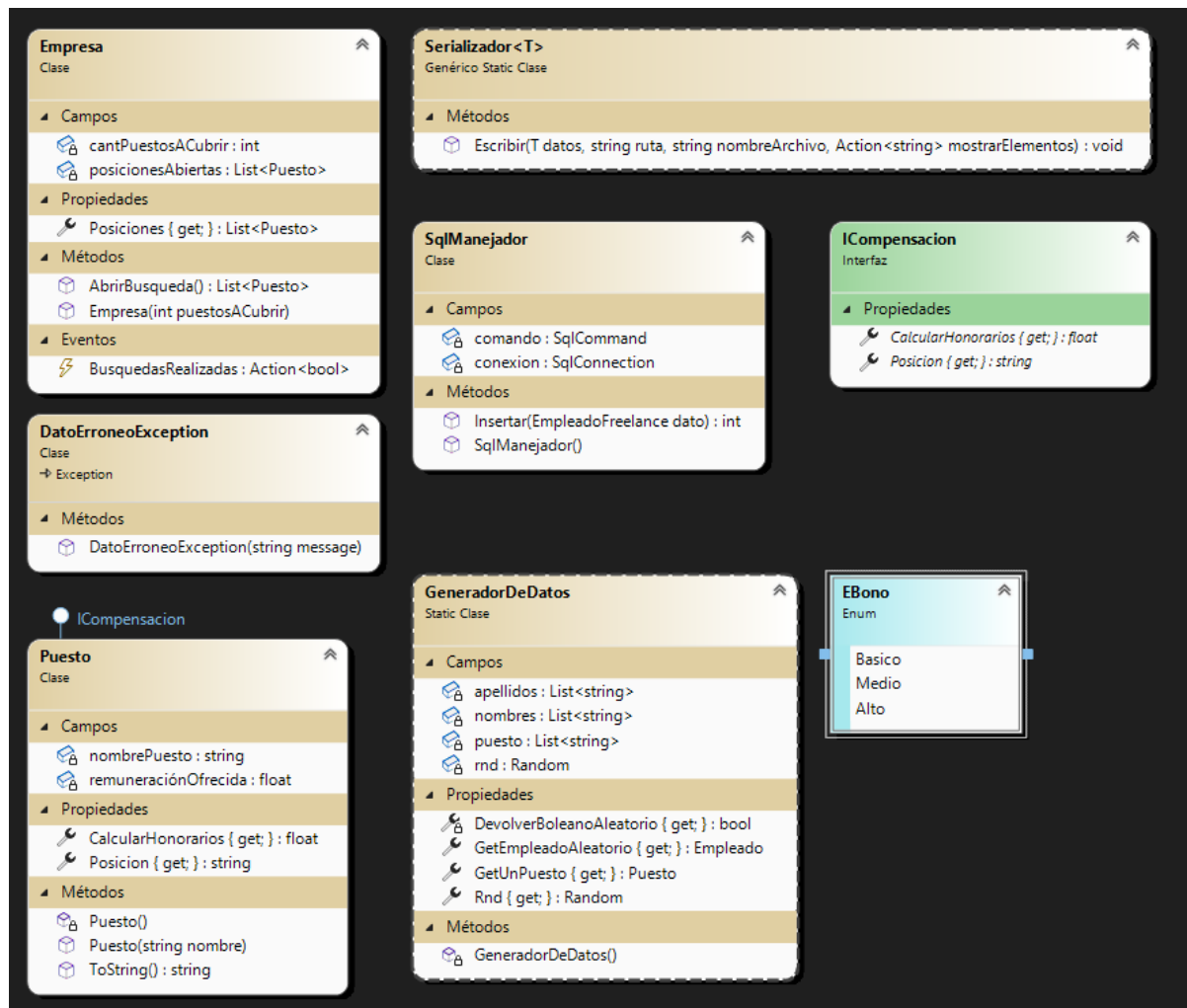
Junto con este documento, recibieron una aplicación a la cual hay que terminar las funcionalidades detalladas más adelante. Tener en consideración que hay funcionalidades ya desarrolladas que **no se deben cambiar**.

De todas formas, si algo les genera duda, pregunten.

Secciones de la aplicación

Biblioteca de clases





Interfaz ICompensacion:

- La interfaz **ICompensacion** deberá implementarse en las clases Empleado y Puesto.
 - En **Empleado**:
 - CalcularHonorarios**: Será abstracto.
 - Posición**: Retorna el atributo Posición.
 - En **EmpleadoFreelance**:
 - CalcularHonorarios**:
 - Si el booleano dolarOficial es **true** la remuneraciónPretendida será convertida a un dólar de 138 pesos.
 - Si el booleano dolarOficial es **False**, será convertida a un dólar de 290 pesos.
 - En **EmpleadoRelacionDependencia**:

- **CalcularHonorarios:**
 - Si la remuneración pretendida excede los 250000, deberá tener un descuento del 30%.
 - Si no excede dicho monto, no tendrá descuentos.
 - **SueldoConAguinaldoActualizado:** A la remuneración Pretendida, le suma un 50% más y luego a eso le suma el valor del tipo de bono que el empleado haya negociado.
- En **Puesto:**
 - **CalcularHonorarios:** Retorna **remuneracionOfrecida**.
 - **Posición:** Retorna el nombre del puesto;

Empleado:

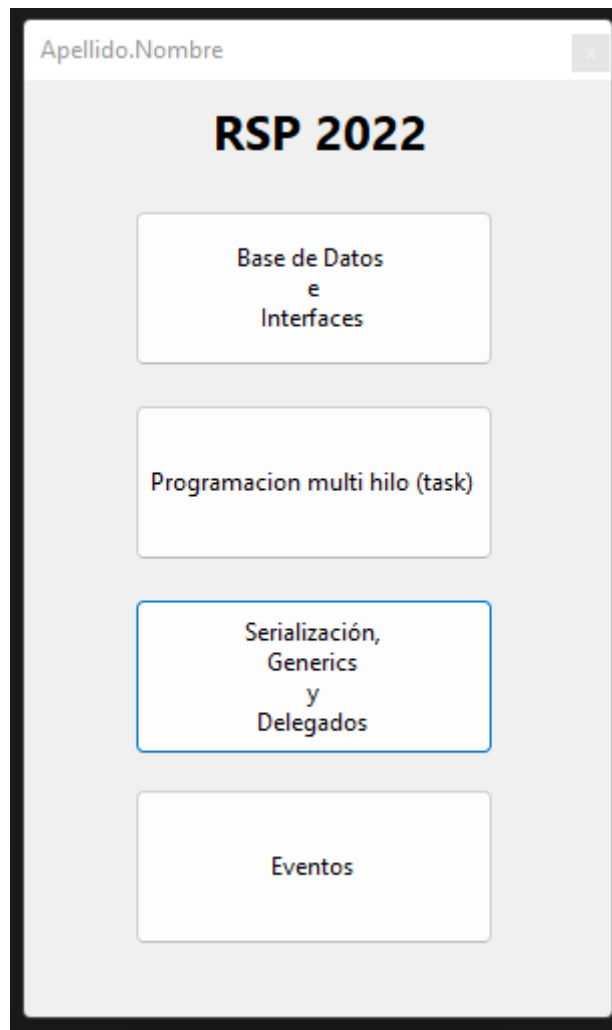
- Su constructor privado le dará un valor de entre 100000 y 250000 al atributo **remuneracionPretendida**, utilizando la propiedad Rnd de la clase **GeneradorDeDatos** (Esto ya está codeado, solo utilizarlo).

Puesto:

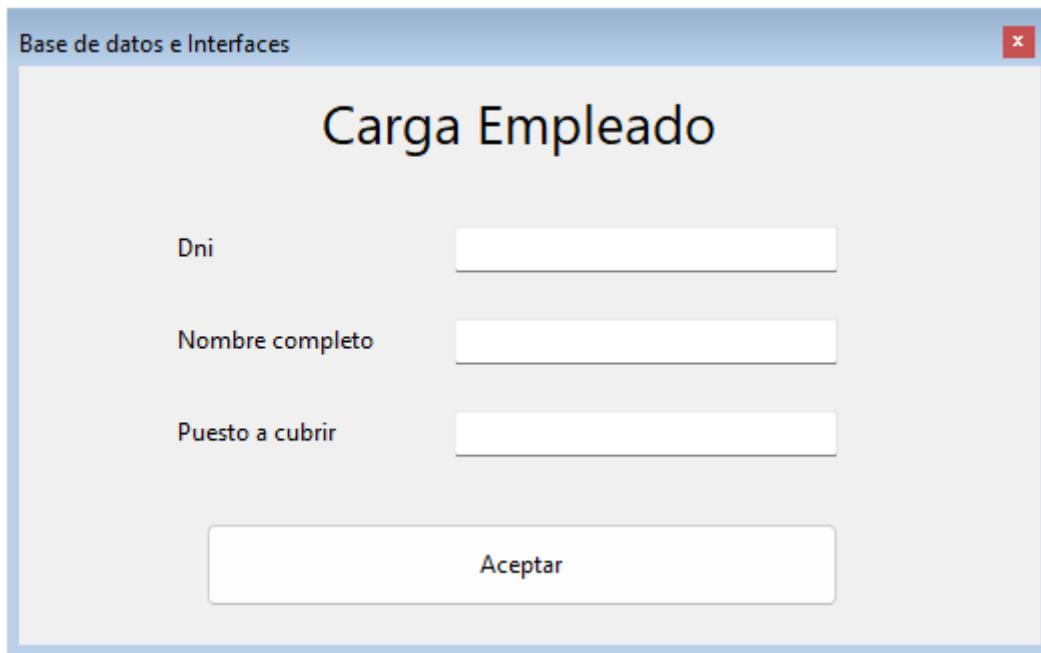
- Su constructor privado le dará un valor de entre 100000 y 300000 al atributo **remuneraciónOfrecida**, utilizando la propiedad Rnd de la clase **GeneradorDeDatos** (Esto ya está codeado, solo utilizarlo).

Menú principal

- En el menú principal de la aplicación solo deben reemplazar el “Apellido.Nombre” por sus propios datos.



Base de datos e Interfaces



Base de datos e Interfaces

Carga Empleado

Dni

Nombre completo

Puesto a cubrir

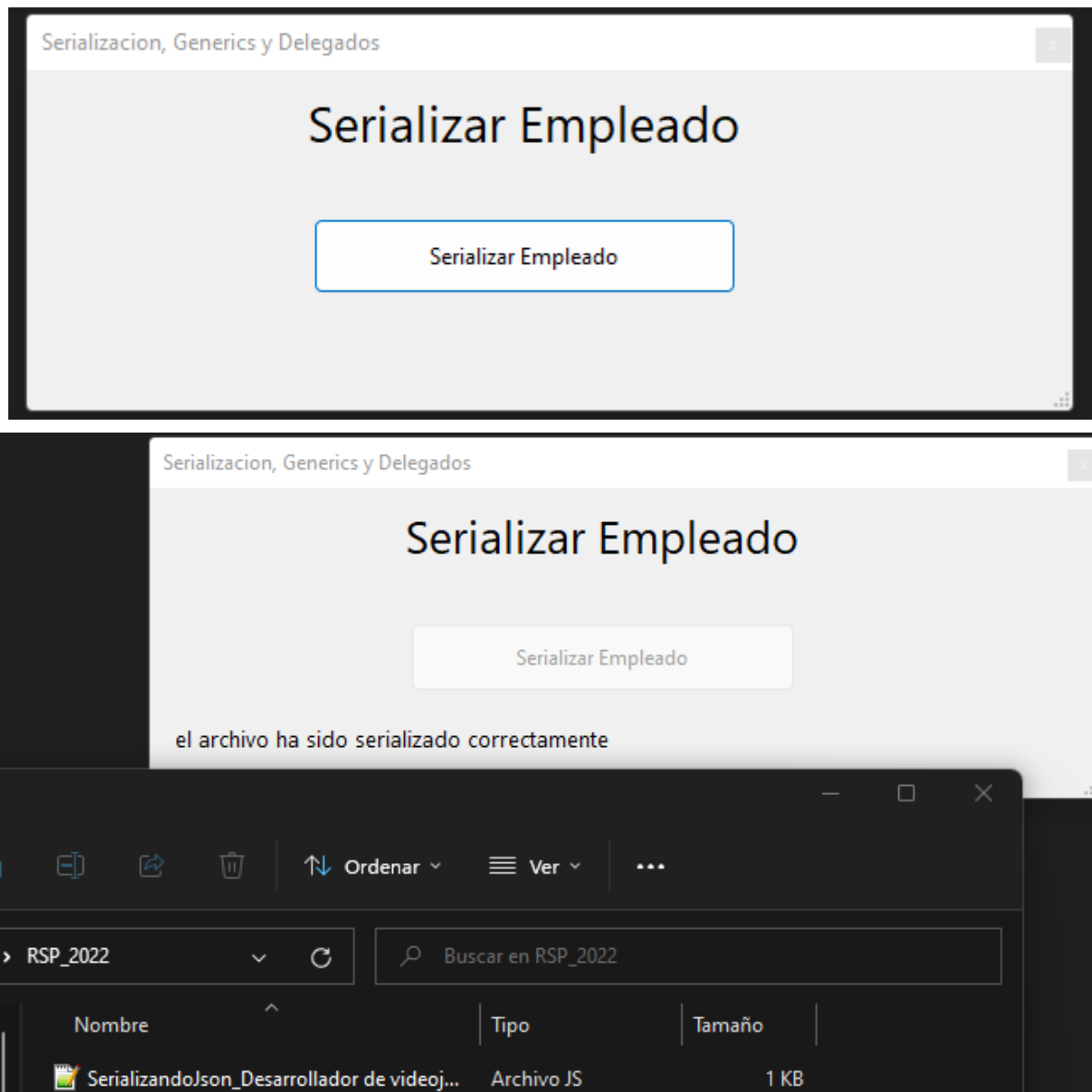
Este módulo deberá insertar un **EmpleadoFreelance** en la base de datos (Al final de este documento encontrarán el script para crear la tabla en la base de datos).

A tener en cuenta:

- Deberán validar que los datos sean correctos. Es decir:
 - Dni: Deberá tener un valor entre 10.000.000 y 45.000.000
 - Nombre: No puede ser ni nulo ni vacío.
- La validación no se hará en el form. Será en el método de **instancia** Insertar de la clase **SqlManejador** (también será de instancia). Si algún valor es incorrecto, se arrojará una excepción de tipo **DatoErroneoException**, que será capturada en formulario mostrando en un messagebox el mensaje.

Serialización, Generics y Delegados

Este módulo lo que hará es serializar un Empleado y mediante un delegado, desactivar el botón para serializar otro empleado y mostrar en pantalla la leyenda “El empleado ha sido serializado correctamente”.



Programación MultiHilos (Task)

El botón Comenzar Liquidacion Sueldos carga correrá en un proceso separado, el método ComenzarCarga. El mismo simulará a cuantos empleados puede contratar dependiendo de sus salarios y el presupuesto de la empresa.

La empresa no puede superar:

- Dólares: No puede superar los 10000 dolares
- Pesos: No puede superar los 10000000 pesos.
- Aguinaldos: No puede superar los 20000000

En otras palabras, dicho método hará el proceso de:

- Obtendrá un Empleado random.
- Lo agregará a la lista de empleados
- Dependiendo de que tipo de empleado, acumulará sus honorarios en el acumulador correspondiente.
- Actualizará la información en pantalla
- Validará no haberse excedido. Si lo hizo, deberá cancelar el proceso.
- Cada nuevo empleado se cargará luego de 2 segundos.

Programación multihilos (Task)

Modulo Liquidacion Empleados

Comenzar Liquidacion Sueldos

A liquidar en dolares:

A liquidar en pesos

Monto de aguinaldos a liquidar:

Programación multihilos (Task)

Modulo Liquidacion Empleados

Posicion	Dni	NombreCompleto	CalcularHonorarios
Lider tecnico	34816422	Ygo,Jarrood	177576
Artista 3D	22161820	Ygo,Jarrood	376,1903
Lider tecnico	17338482	Braniff,Elena	354,52942
Artista 3D	35671799		
Diseñador UX/UI	13069128		

Hasta aqui dan las finanzas: Se les pagará a 10 empleados

Comentarios

Aceptar

Sueldos Dolarizados a Liquidar mensualmente: 5351,243

Sueldos Pesificados a Liquidar mensualmente: 554204

Monto a liquidar en Aguinaldos: 906306

Eventos

Se presionará el botón “Publicar Puesto Trabajo” hasta que el cupo de postulaciones de la empresa esté lleno.

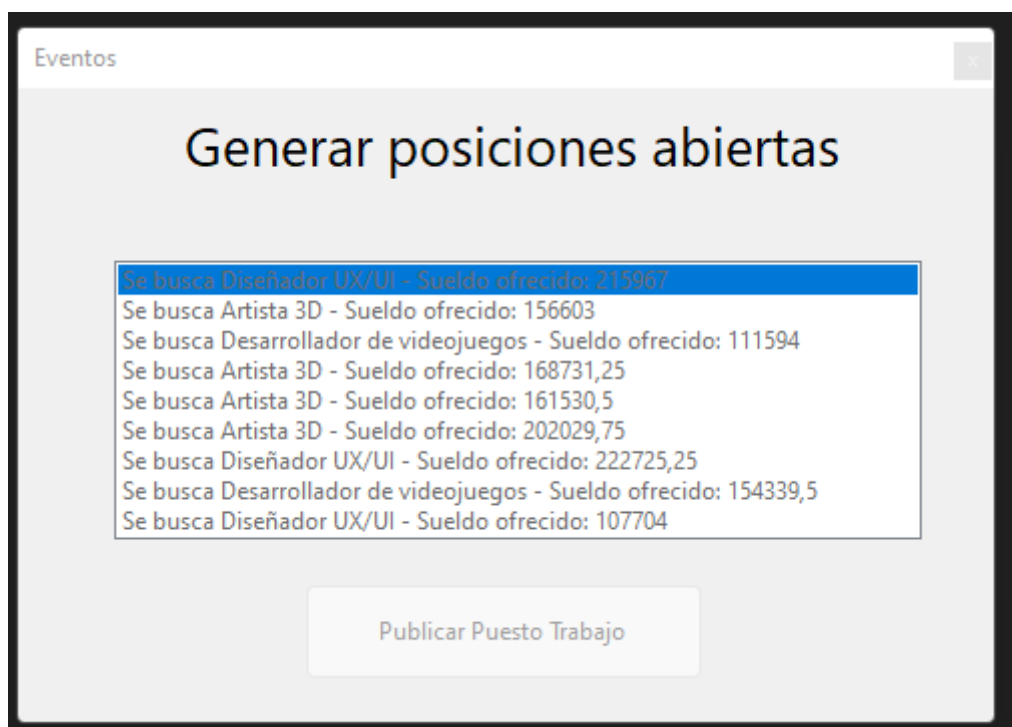
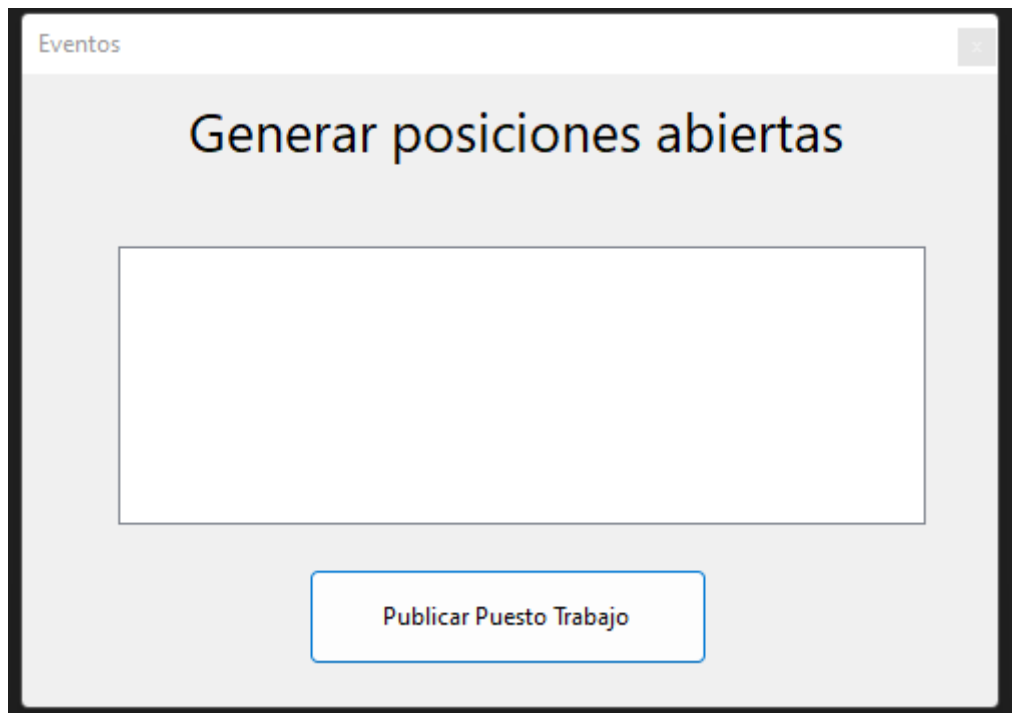
Al llenarse, se deshabilitarán los componentes y se mostrará por pantalla el siguiente mensaje en un messagebox:

“Todas las postulaciones han sido realizadas con éxito.

Agradecemos su participacion.

Los esperamos la proxima”.

*Tip: La propiedad **GetUnPuesto** de la clase estática **GeneradorDeDatos** devuelve un Puesto aleatorio.*



Test Unitarios

Se piden **dos** de los siguientes casos:

- Testear que al insertar un Empleado con valores incorrectos, se arroja correctamente la excepción que corresponde.
- Testear que un Empleado se insertó correctamente. (Tip: El método `ExecuteNonQuery()` devuelve el número de rows insertadas, por lo tanto si es mayor a 0, quiere decir que se insertó correctamente).
- Testear que los métodos **GetUnEmpleado** y **GetUnPuesto** devuelven datos válidos (En el mismo unit testing).

Script Base de datos:

- 1) Crear una base de datos llamada **ExamenPrimerFecha2022**
- 2) Correr el siguiente script

```
USE [ExamenPrimerFecha2022]
GO
/***** Object: Table [dbo].[Empleados]    Script Date: 1/8/2022 19:44:49 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Empleados](
    [Dni] [int] NOT NULL,
    [Nombre] [nvarchar](50) NOT NULL,
    [Posicion] [nvarchar](50) NOT NULL,
    [Honorario] [int] NOT NULL
) ON [PRIMARY]
GO
```