



# PROGRAMACIÓN ORIENTADA A OBJETOS

## TRABAJO PRÁCTICO N°4

1. Desarrolle dos ejemplos del uso de un dialog:
  - a. Un cuadro de diálogo que indique el siguiente mensaje “Está seguro que quiere dar de baja al usuario”, con los botones de aceptar y cancelar.
  - b. Un cuadro de diálogo que sea capaz de obtener un texto ingresado por el usuario y que luego lo muestre en otro cuadro de diálogo.
2. Crear una interfaz gráfica de usuario (IGU) que permita seleccionar un archivo y una ubicación de destino. Luego, al presionar un botón este archivo debe almacenarse en la ubicación indicada.
3. Defina una IGU para el formateo de datos. Esta debe poseer los métodos:
  - public static String formatearMoneda(Double cantidad)
  - public static String formatearFecha(Integer dia,Integer mes,Integer año)

Luego cree dos implementaciones de esta interfaz

- Primera
  - Moneda en formato estadounidense
  - Fecha en formato estadounidense
- Segunda
  - Moneda en formato argentino
  - Fecha en formato argentino

Finalmente, cree una interfaz gráfica que permita ingresar

- Una cantidad de dinero
- Un día, un mes y un año(tres campos)
- Mediante un radio button, el formato elegido de entre los dos posibles.

Luego, al presionar un botón debe completarse un label con la cantidad de dinero formateada y otro con la fecha formateada.

En el caso de que el día o mes no sean correctos debe arrojar una excepción creada por usted. La excepción debe capturarse y mostrar un mensaje en pantalla mediante un cuadro de diálogo.

4. Defina la interface idioma con los siguientes métodos:
  - saludar().
  - despedirse().
  - perdon().
  - pedirCafe().
  - cuantoCuesta().



## PROGRAMACIÓN ORIENTADA A OBJETOS

### TRABAJO PRÁCTICO N°4

- dondeQueda().

Luego programe las clases Ingles, Frances y Portugues que van a implementar esta interface y van a devolver el mensaje en el idioma que corresponda en un objeto de tipo String.

Cree una IGU que permita seleccionar el idioma y el mensaje para poder ver su traducción correspondiente.

5. Desarrolle una calculadora que permita:
  - Ingresar números enteros y con decimales
  - Mostrar la operación completa que se está realizando actualmente(Un display)
  - Borrar la operación completa que se está realizando(Reiniciar display)
  - Sumar, restar, multiplicar y dividir dos números ingresados
  - Debe capturarse la excepción y mostrar un mensaje al usuario cuando se intente dividir por cero.
6. Crear un nuevo proyecto que incluya las clases Empresa, Persona y Puesto resultantes del ejercicio 13 del práctico 2.

En el inicio del programa, cree un listado de Empresas y desarrolle una IGU que las muestre en una tabla.

7. A partir de lo realizado en el ejercicio anterior añada la siguiente funcionalidad
  - La tabla debe permitir seleccionar una fila y solo una fila a la vez.
  - La pantalla debe contar con un botón Eliminar por encima de la tabla. El mismo debe estar deshabilitado por defecto.
  - Cuando se selecciona una fila se debe habilitar el botón Eliminar.
  - Al presionar el botón Eliminar se debe eliminar la fila seleccionada de la tabla y la empresa correspondiente del listado creado al inicio del programa.
  - Luego de eliminar la empresa el botón debe volver a quedar deshabilitado hasta que se seleccione una nueva fila.
8. (Este ejercicio requiere una instalación de PostgreSQL)

Desarrolle una IGU que permita ingresar los datos de conexión a una base de datos postgres y realice la implementación necesaria para realizar la conexión utilizando esos datos

- Los campos que debe poseer la interfaz son
  - Host, Puerto, Base de datos, Usuario, Contraseña



# PROGRAMACIÓN ORIENTADA A OBJETOS

## TRABAJO PRÁCTICO N°4

The image shows a window titled 'Test conexión' with a standard Windows title bar (minimize, maximize, close buttons). Inside the window, there are five text input fields stacked vertically, each with a label above it: 'Host', 'Puerto', 'Base', 'Usuario', and 'Password'. Below these fields is a single button labeled 'Probar conexión'.

- Luego, debe haber un botón que permita probar la conexión hacia dicha base de datos
    - En caso de que la conexión sea exitosa debe mostrar un mensaje de éxito al usuario
    - En caso de que los datos brindados no permitan realizar la conexión tiene que mostrar un mensaje de error
9. Crear una IGU que permita seleccionar un archivo en formato CSV con el siguiente formato de línea.

Nombre,apellido,fecha de nacimiento(DD/MM/YYYY), edad, dni, localidad

Luego, al presionar un botón debe leerse el contenido de dicho archivo y mostrar, por cada línea, algunos valores en una tabla. Dicho valores serán: Nombre, apellido y dni.

En caso de que el archivo no se encuentre en formato CSV debe despacharse una excepción.

En caso de que alguna línea del archivo no cumpla con el formato indicado debe despacharse una excepción.

Ante cualquier excepción debe abortarse el procesamiento del archivo y mostrar un mensaje en pantalla.



## PROGRAMACIÓN ORIENTADA A OBJETOS

### TRABAJO PRÁCTICO N°4

10. Realice un juego de memoria con colores.

La pantalla debe tener un botón para iniciar partida. Al presionarlo se deben mostrar doce tarjetas de las cuales habrá dos por cada color, siendo seis el total de colores. Luego de 15 segundos se deben ocultar los colores de las tarjetas, mostrando para todas un color en común que no se haya utilizado.

En la parte superior debe mostrarse un contador de segundos que inicia al ocultarse los colores de las tarjetas.

El jugador deberá hacer click en dos tarjetas y en cada click la tarjeta mostrará su color:

- Si los colores coinciden ambas tarjetas quedan visibles.
- Si no coinciden los colores deben volver a ocultarse los colores

El juego termina cuando todas las tarjetas quedaron al descubierto, es decir, cuando los seis pares de colores se encontraron. En ese momento debe mostrarse un mensaje de felicitaciones al jugador y el tiempo total que le llevó.

