

# Trabajo Práctico 3

## Programación III

Integrantes: Córdoba, Deborah | Gaona, Guillermo

En este informe explicaremos la implementación del código de la aplicación y el diagrama de clases.

La aplicación cuenta con 7 clases:

1. Álbum
2. Interfaz
3. Figuritas
4. Paquete
5. Simulacion
6. Escenario 1
7. Escenario 2
8. Escenario 3

### **ÁLBUM**

Clase encargada de crear el álbum, con su cantidad total de figuritas disponibles para pegar, dentro de ella podemos encontrar las funciones

getCantidadFigs: devuelve la cantidad de figuritas del álbum.

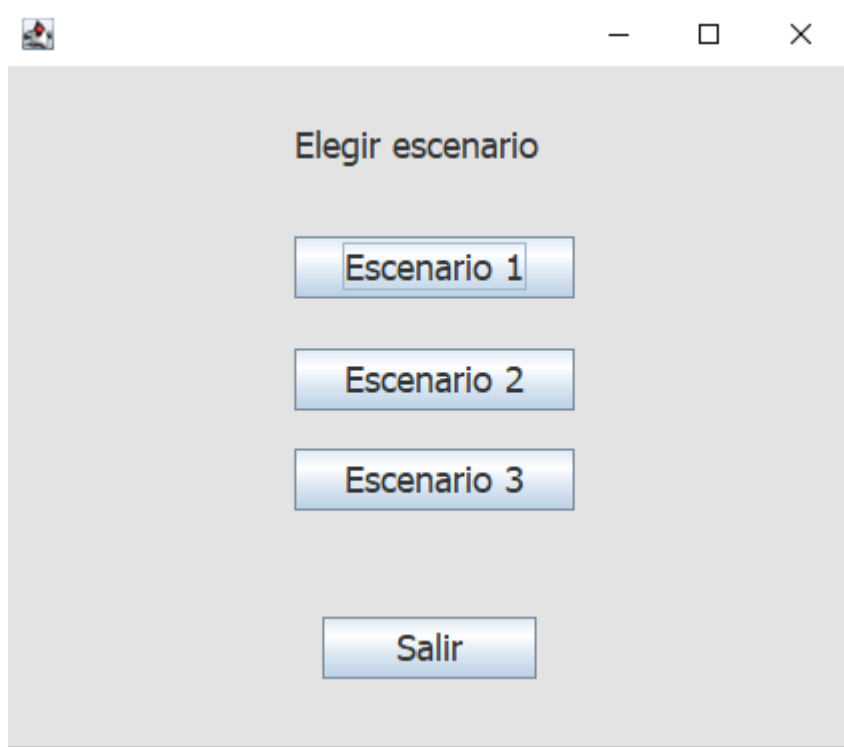
tamañoAlbum: devuelve la cantidad de figuritas pegadas en el álbum.

getFiguritas: devuelve las figuritas que tenemos pegadas en el álbum.

agregarFigu: recibe como parámetro un entero siendo este la figurita y la agrega a las figuritas del album.

### **INTERFAZ**

Clase encargada de mostrar la parte gráfica de la aplicación y crear las simulaciones.



## **FIGURITAS**

En esta clase vamos a tener un stock de figuritas por cada figurita

Figuritas:constructor de figuritas donde se declara la cantidad de figuritas por cada jugador,y la cantidad total de figuritas.Dentro de ella podemos encontrar las funciones.

getFiguritas:retorna las figuritas.

revisarStock:recibe como parámetro un jugador y retorna la cantidad de stock de ese jugador.

tamaño:retorna el tamaño de las figuritas.

disminuirStock:recibe como parámetro el número de jugador y le reduce -1 en su stock .

## **PAQUETE**

En esta clase vamos a contener 5 figuritas por cada paquete.Dentro de ella podemos encontrar las funciones.

Paquete:constructor de paquete.

cargarPaquete:esta función se encarga del llenado de paquetes con figuritas aleatoriamente seleccionadas del total de figuritas, cada vez que agrega una revisa que no esté repetida en el sobre y disminuye su stock en el total de las figuritas.

getFigu:recibe como parámetro el número de una figurita y retorna esa figurita del total de figuritas.

tamañoPaquete:devuelve el tamaño del paquete.

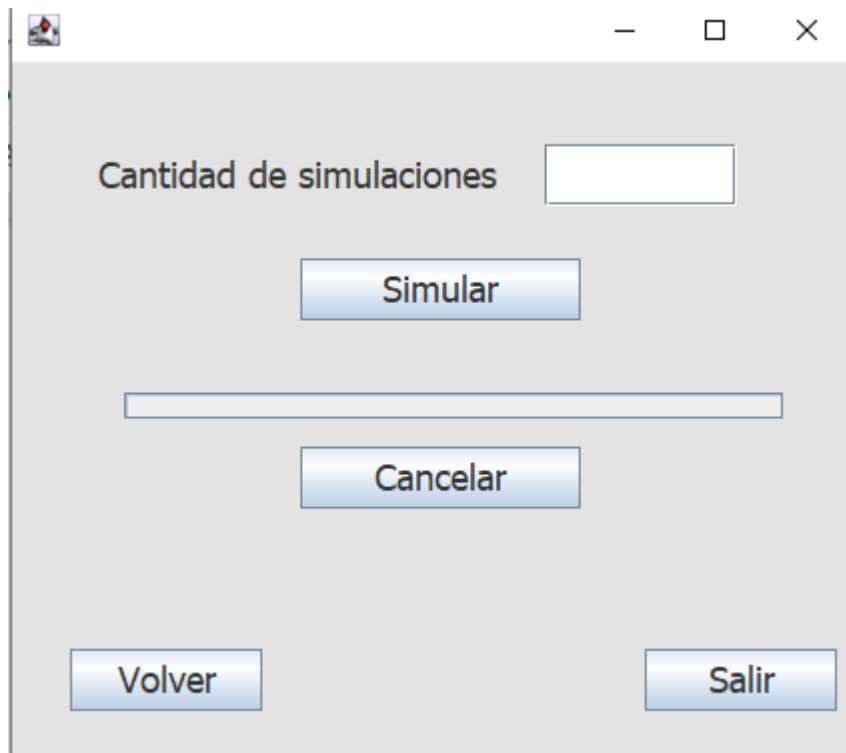
### **SIMULACIÓN**

Acá podremos encontrar las distintas opciones de escenarios dependiendo el criterio que se quiere para hacer la simulacion del llenado de álbumes, en el caso de escenario2 y escenario 3 se va a crear un thread por cada simulación, dentro de cada escenario al término de todas las simulaciones se mostrará un promedio de la cantidad de paquetes necesarios para llenar un álbum que tiene de tamaño 638 figuritas. Y graficar el progreso mediante una barra de progreso.

### **ESCENARIO1**

En esta clase vamos a tener en cuenta la cantidad de simulaciones y un solo usuario llenando el album, dentro de la misma vamos a tener las siguientes funciones

llenarAlbum:en esta función se crea un nuevo álbum y se van abriendo paquetes hasta completar el álbum y va a retornar la cantidad de paquetes que se abrieron para completarlo.



## **ESCENARIO2**

Escenario2:Constructor de la clase , recibe como parámetro figuritas,y la cantidad de usuarios,dentro de ella podemos encontrar las funciones.

run: así como en el escenario 1 va a estar el completado del álbum, creando un nuevo álbum que se va a ir llenando a medida que se vayan abriendo los paquetes, a diferencia del Escenario1 la apertura de paquetes va a ir por “turnos” y en el caso de que un usuario tenga la figurita repetida la va a guardar en un ArrayList que va a contener las repetidas de los usuarios, estos van a poder ver si tienen alguna que le falta y agarrarla para sus propios álbumes.

GetPaquetes:Devuelve la cantidad de paquetes en total que se abrieron.

Cantidad de simulaciones

Cantidad de usuarios

Simular

Cancelar

El promedio es: 000.00

Volver Salir

### **ESCENARIO3**

Escenario3: constructor de escenario 3 que recibe como parámetro la cantidad de usuarios, en esta clase el arreglo de repetidas va a ser individual para cada usuario a diferencia del escenario 2. Dentro de ella podemos encontrar las funciones.

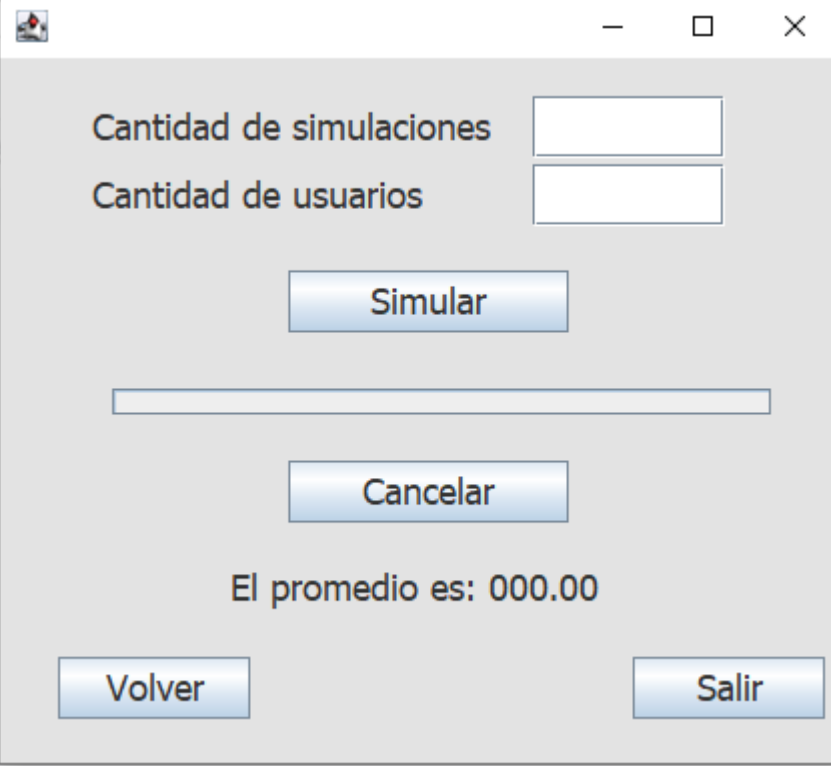
run: encargado de crear y llenar los álbumes de cada usuario, a diferencia del escenario 2 para poder agarrar una figurita repetida el usuario va a tener que tener el en su ArrayList de repetidas una que el otro usuario no tenga en su álbum y así poder intercambiar entre esos dos.

intercambiar: esta función recibe como parámetro un valor int siendo este el identificador del usuario en el ArrayList de usuarios, y un int de figurita siendo este el valor de la figurita que va a querer intercambiar, esta función va a revisar dentro de los álbumes de los otros usuarios a ver a quien le falta, una vez que encuentra uno va a revisar en sus repetidas a ver si tiene una figurita que le sirva al usuario ingresado por parámetro en el caso de intercambiar agrega cada figurita en los álbumes de dichos usuarios, borra la figurita repetida de la lista y retorna true, caso contrario que a nadie le sirva retorna false.

EstadoAlbumes: recibe como parámetro la cantidad de usuarios y va a revisar la cantidad total de figuritas que tienen en los álbumes, si la cantidad total de figuritas

es igual a la cantidad total del figuritas que se pueden pegar en un álbum por la cantidad de usuarios retorna true, caso contrario false.

getCantPaquetesUsados:retorna la cantidad de paquetes usados.



Simulador

Cantidad de simulaciones

Cantidad de usuarios

Simular

Cancelar

El promedio es: 000.00

Volver Salir

