



INFORME PROYECTO BINGO

26/02/2016

Programa desarrollado por estudiantes de informática.

TABLA DE CONTENIDO

Contenido

PLANTEAMIENTO DEL PROBLEMA	1
ANALISIS DEL PROBLEMA	3
acerca del proyecto	3
estrategia de tdd	3
entorno de desarrollo	3
herramienta de java	¡Error! Marcador no definido.
Informe de librerías utilizadas	3
metodologia de desarrollo	3
ESTRATEGIA DE SOLUCION	4
Pruebas	4
Diseño	6
modelo del programa	6
DIAGRAMA DE CLASES	7
Codificación	8
Refactorización	9

PLANTEAMIENTO DEL PROBLEMA

En el presente trabajo se ha de desarrollar el juego del bingo en su versión clásica con 90 bolillos, para ello se requiere el nombre del jugador, un bombo donde se introducen los bolillos, el cartón donde el jugador procederá a marcar los números que salgan del bombo, ganara si el jugador si marca la totalidad de los números de su cartón y para ello deberá gritar bingo, donde la mesa o jurado procederá a la verificación de las marcaciones. Si esta afirmación es correcta habrá concluido el juego.

Para el desarrollo del mismo se han detectado las siguientes funcionalidades:

1. Generar cartones.-

Beneficio: permitirá proporcionar el carton al jugador para que realicen las marcaciones respectivas.

Criterios de aceptación: para la generación de cartones, primero los mismos deberán tener las siguientes características:

- formato 3 filas 9 columnas,
- cada fila debe contener solo 5 números que se encuentren en el rango del 1 al 90 incluyendo estos números.
- No se Deben repetir los números en una tarjeta

F1	F2	F3	F4	F5	F6	F7	F8	F9
20	***	25	***	***	27	***	49	60
9	35	***	***	***	42	50	65	***
12	26	***	30	33	***	***	***	65

Como se puede apreciar en el cartón existe siempre 4 espacios vacíos por fila (**); según establecen las reglas del juego, asimismo 5 datos de valores positivos <=90.

2. Un bombo o tómbola.-

Beneficio: permitirá iniciar el juego a través de la obtención de bolillos para que se marquen los cartones.

Criterios de aceptación:

- deberá contener 90 bolillos del 1 al 90 incluyendo estos números,
- una vez se saque el bolillo el mismo deberá leerse y mostrarse al jugador para no recibir ninguna queja.
- Los bolillos solo salen uno a la vez.
- La extracción de bolillos termina cuando algún jugador gana.

3. Marcar los cartones.-

Beneficio: Para que el jugador sea capaz de marcar los cartones.

Criterios de aceptación: El marcado debe realizarse de tal forma que permita:

- verificar si el marcado se realizó correctamente.
- Así mismo una vez se marquen todos los números del cartón se debe gritar bingo para anunciar



si se ha ganado. Y con ello pasar a la verificación.

4. Verificar los cartones marcados.-

Beneficio: Permite detectar errores en el marcado anunciar o rechazar a la persona que grita bingo.

Criterios de aceptación:

- efectuara la comparación de los números marcados con los números que se han obtenido en la tómbola,
- cuando las coincidencias se realicen en los 15 números anunciara al ganador y finaliza el juego, caso contrario devuelve el cartón y prosigue el juego.

ANALISIS DEL PROBLEMA

ACERCA DEL PROYECTO

La idea es desarrollar un programa capaz de jugar bingo con todas sus modalidades de juego.

Dado el hecho de que el juego de bingo se realizara de forma virtual cambiamos el aspecto de jugar pero mantendrá las reglas del BINGO original

ESTRATEGIA DE TDD

Se ha de implementar la metodología de TDD, para ello se desarrollaran las pruebas previamente de forma que permitan verificar el cumplimiento de las funcionalidades que se desarrollaron en el titulo anterior.

ENTORNO DE DESARROLLO

- Netbeans: es un entorno de desarrollo integro libre, echo principalmente para el lenguaje Java.
Un entorno muy completo y profesional para la iteración de usuarios.

Informe de librerías utilizadas

Importaciones que provienen del package de import Java. Útil.*; y Import Javax. Swing.*;

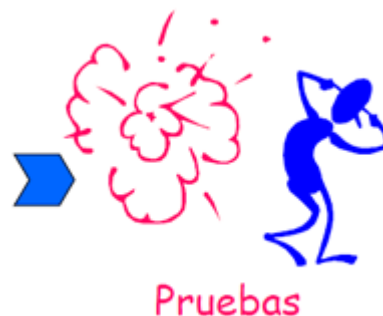
METODOLOGIA DE DESARROLLO

- P.O.O (programación Orientada a Objetos).
- Herencia.
- Métodos privados, públicos.
- Hilos
- Creación de objetos dentro de otros objetos

ESTRATEGIA DE SOLUCION

Pruebas

Para el diseño de pruebas debemos que indagar en las funcionalidades que se han detectado, en nuestro caso cuatro funcionalidades, en función a los criterios de aceptación de estas funcionalidades sistematizaremos las pruebas necesarias, ya que posteriormente cuando diseñemos el código nos permitan cerciorarnos de que el programa realice correctamente las tareas que programamos.




Para este objeto se han diseñado las siguientes pruebas: En este caso nos desenvolveremos en el uso de tres clases que se serán las que ejecuten las funcionalidades:

1. Clase generador de tablas.- La clase se ocupara de ejecutar las funcionalidades de generar los cartones con las condiciones requeridas, Para esta clase se han de diseñar las siguientes pruebas:
 - Test que verifique que solo existan 4 espacios vacíos representados en este caso con "***"
 - Test que verifique que no existan repetidos dado un determinado cartón
 - test que verifique que los datos de la tabla sean mayores a 0 y menores que 91
2. clase validador.- Esta clase se ocupara de verificar que los números marcados correspondan a los que han salido en el bombo o tómbola, una verificado sea el mismo procederá a marcarlo con una X, en caso de que el jugador intente marcar un número que no haya salido en la tómbola no permitirá el marcado. En este se desarrollara un test que permita verificar si dado un numero que proporcione la tómbola y dad una posición para marcar, se verifique la igualdad de los mismos, en este caso se marque con "X"
3. Clase jurado.- esta clase contendrá primero un método que verifique que la cantidad de casillas marcada sea igual a 15 cuando el jugador grite bingo, el verificador de que no salgan números repetido en la tómbola, así mismo también el método q finalice el juego una vez se culminen el número de intentos Para ella desarrollaremos los siguientes test:
 - Un test que verifique la cantidad de acierto sea igual a 15
 - un test que verifique que los números de la tómbola no se repiten.

Adicionalmente se ha de diseñar una interface con el usuario que permita desarrollar el juego de acuerdo a las estipulaciones mencionadas de forma previa.

Por lo mencionado anteriormente se ha de hacer pruebas para cada funcionalidad, esto significa que cada método, clase, panel o pantalla visualizada deberá previamente pasar los test prediseñados, por lo tanto garantizara que los mismo sean realizados sin errores. Con ello nos cerciorarnos que esta sección corre correctamente.

Para el tema de la interface se realizaran las pruebas como usuario final



Después de diseñado el código y de verificar que los test están positivos, podremos llegar al producto a la conclusión de que el programa corre sin errores.

Una vez que el programa corre con total normalidad debemos eliminar los BAD SMELL respectivos (para un buen funcionamiento del proyecto)

Diseño

MODELO DEL PROGRAMA

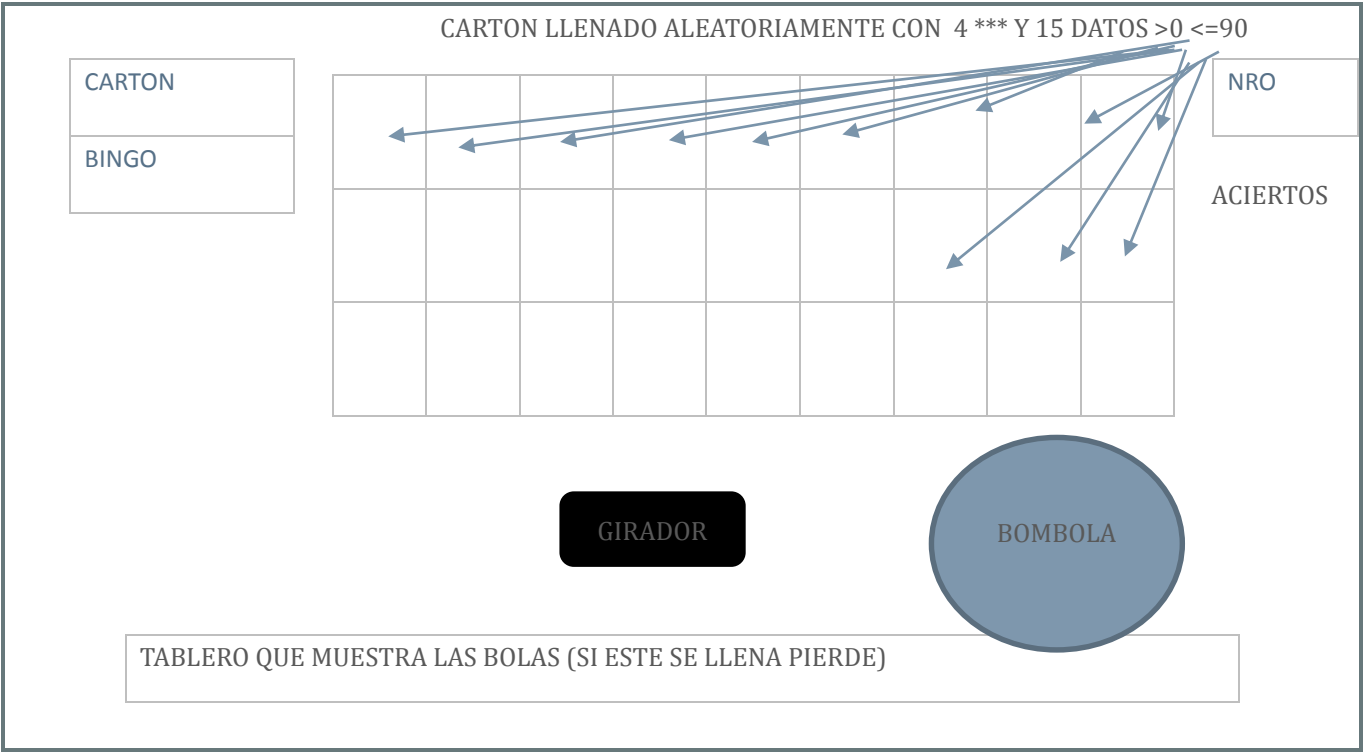
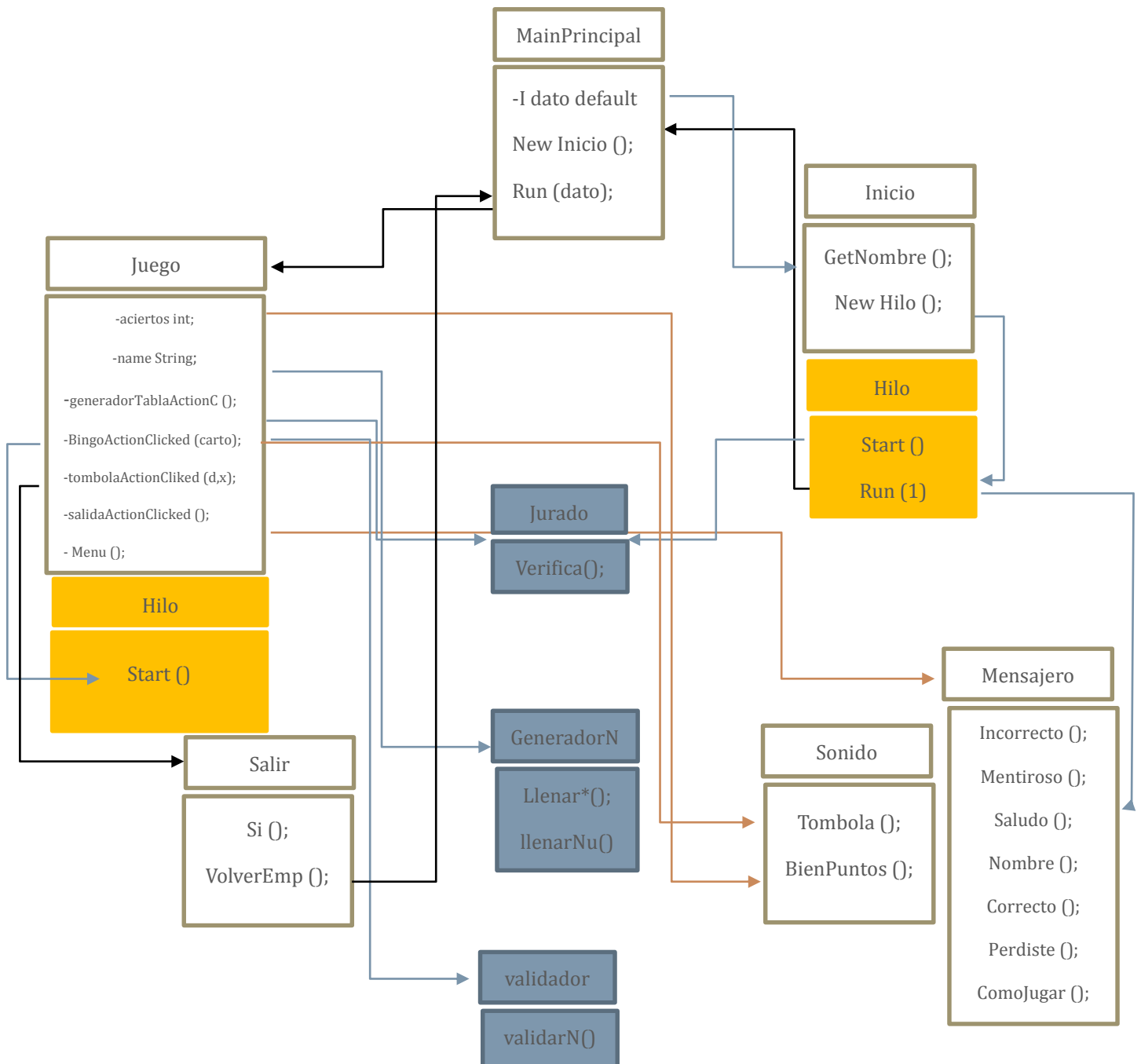


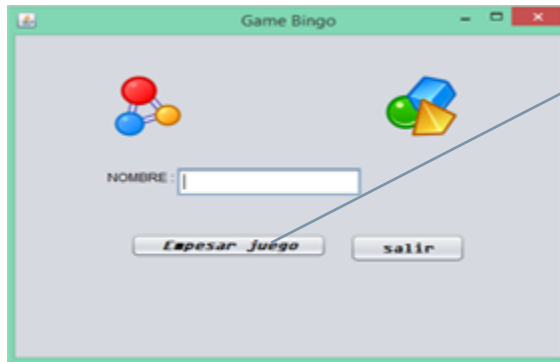
DIAGRAMA DE CLASES



Codificación

Existe un Main que controla cada Panel especie de hilo multiHilo;

Vista controlador: creamos un JPanel donde nos creamos objetos JLabel y JBoton para acceder e ingresar valores; llamamos al metodo getText(); para obtener el nombre del usuario de un TextField



La clave de acceso está aquí ya que controla

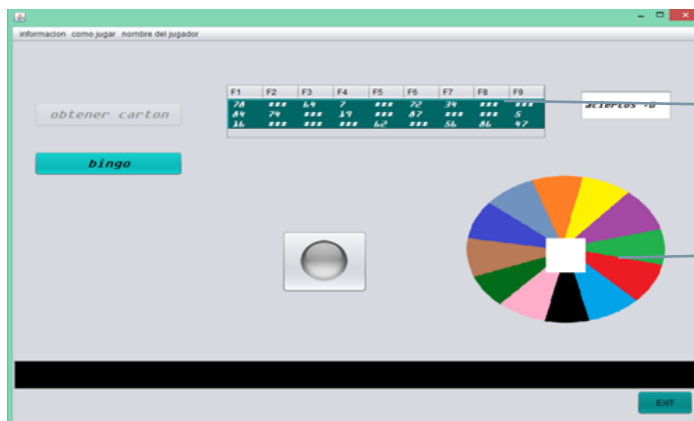
La cadena ingresada si es correcto.

Nace el hilo para generar un cargando, al mismo tiempo el programa mandara un saludo obteniendo el nombre

Posteriormente dará la indicación al Main para que continua con el siguiente Panel, solo así abrirá el siguiente caso de no ser incorrecto mostrara un mensaje “incorrecto”.

Este es el bingo en la parte superior existe un menú de opciones que da la información acerca del usuario....la tómbola de igual forma está en un Hilo porque genera una serie de imágenes y por su parte números... es la cuestión por lo usa un hilo que extiende de la Thread.

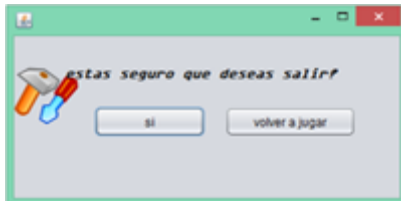
El cartón es obtenido a través de un JBoton que manda a una clase (generar Tabla) para realizar dicha operación el tablero mostrado es JTable el botón plateado ,que genera la aleatoriedad del numero una vez mandada al método generadorActionClicked ();



$n > 0 \ \&\& \ n \leq 90$

$n > 0 \ \&\& \ n \leq 90$

Este Panel se habilita una vez que presionas el botón exit, de igual forma da la alarma al Main para que pueda activarte esta ventana, creando dos opciones, la opción si: System.exit (si es que existiera un error seria 1, de otro modo 0); la opción volver => envía los datos al principal y este reconfigura el juego manteniendo el nombre y volver al juego.



Refactorización

Para poder hacer una refactorización debemos entender primero qué es eso y para qué nos sirve: "según la definición son los Cambios que se realizan a la estructura interna del software para facilitar la comprensión y abaratar el mantenimiento sin cambiar su comportamiento

Una vez re factorizado comenzamos con la documentación respectiva.

