

NOMBRE		CICLO FORMATIVO	CURSO
		DAW/DAM	1º
APELLIDOS		MODULO	CONVOCATORIA
		PROGRAMACIÓN	EXTRAORDINARIA
DNI	FECHA	NOTA	
	23 DE JUNIO DE 2023		

IMPORTANTE: CADA ENTREGABLE DEBE COMPILAR PARA SER CORREGIDO

Una empresa dedicada a los juegos de azar desea desarrollar una aplicación para sus locales. Para tal fin desean empezar por diseñar una aplicación para el bingo. El bingo, contendrá diferentes bolas, que además de un número, se quiere que se muestren con un color, que puede variar según desee el dueño del local.

En el bingo, podrán jugar un número ilimitado (solamente limitado por los recursos del equipo informático) de jugadores, donde cada jugador dispondrá de un numero de cartones sin determinar. Se desea desarrollar la aplicación en base a las siguientes restricciones:

- La resolución del problema se ha de realizar obligatoriamente siguiendo una programación orientada a objetos.**
- Los jugadores, independientemente de si juegan al bingo o a otro juego, dispondrán de un nombre, apellidos y un saldo. Los jugadores de bingo dispondrán de una cantidad ilimitada de cartones diferentes.
- Se desea diseñar un bombo de tal manera que pueda ser utilizado tanto para bolas de bingo como para en un futuro albergar bolas de lotería u otro tipo de objeto todavía por definir.



- Se ha de diseñar de tal manera que un bombo no pueda contener 2 bolas iguales, entendiendo por bolas de bingo iguales, aquellas que tienen el mismo número independientemente de su color. **Para evitar este problema, se debe desarrollar el bombo utilizando la colección Set.**
- Los bombos utilizados para el bingo, cada 5 bolas extraídas, ha de girar para mover las bolas. Este comportamiento debe ser automático.

NOMBRE	APELLIDOS

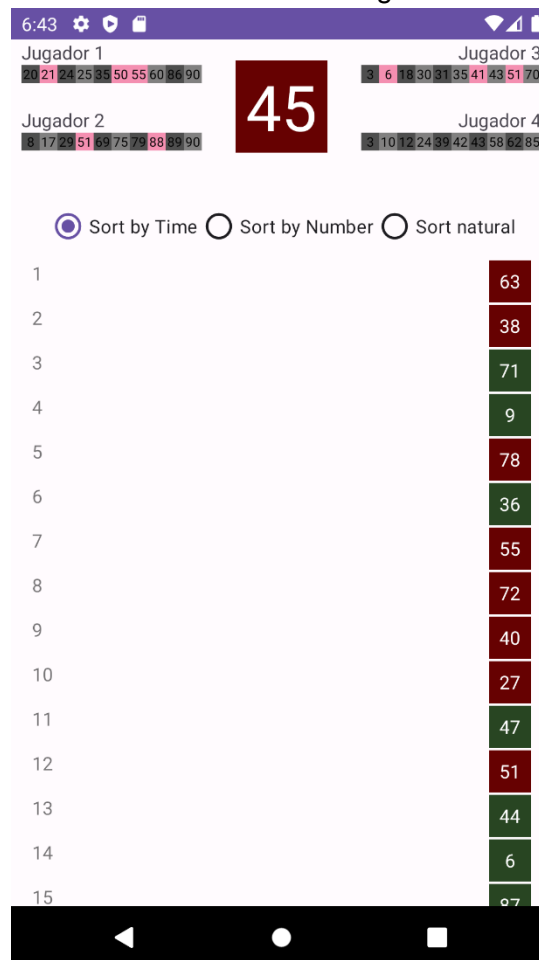
- f. Los cartones contendrán 10 números que pueden ir desde el 1 al 90 sin repetición y ordenados. **Para evitar este problema, se debe desarrollar el cartón utilizando la colección Set.**

Cartón N° 79 www.bingosocial.es · 645 84 05 16 Bingo Social

3		22		41	52			89
7	12		38		57		73	
	18	25		44		64	76	

Para poder comprobar el funcionamiento del sistema, se desea crear una App con las siguientes características:

- Diseña una app con una interfaz similar a la siguiente:



- La app tendrá 3 partes:

NOMBRE	APELLIDOS

- Se debe visualizar la bola que sale del bombo. deberá mostrar tanto el numero como el color. Para volver a sacar otra bola, bastara con hacer clic sobre el TextView.

45

- Se dispondrán de una serie de RadioButtons para ordenar el RecyclerView de tres maneras diferentes:

☒ Sort by Time
 ☐ Sort by Number
 ☐ Sort natural

- **Sort by Time:** El RecyclerView mostrará las diferentes bolas según el orden en el que han ido saliendo
- **Sort by Number:** El RecyclerView mostrará las bolas ordenadas por número.
- **Sort natural:** El RecyclerView mostrará las bolas ordenadas por su orden natural, este es: primero por color y ante dos bolas del mismo color, por número.

- Se dispondrá de un RecyclerView, donde podremos ver las bolas que han salido. Estas deberán ordenarse cuando el usuario haga clic en los RadioButtons anteriores:

1	63
2	38
3	71
4	9
5	78
6	36
7	55
8	72
9	40
10	27
11	47
12	51
13	44
14	6
15	23

- Se pide hacer una simulación para 4 jugadores. Aunque las clases puedan utilizar más de un cartón y deban estar desarrolladas para ello, en la app, para simplificar solo se podrá jugar con un cartón.
- Cada vez que arranque el juego, se deberá borrar todos los datos de la BBDD. Esto se deberá realizar utilizando la clase CallableStatement.

NOMBRE	APELLIDOS

- A medida que el juego vaya discurriendo, cada vez que se obtenga una bola del bombo, se debe:
 1. Mostrar la bola en el display
 2. Añadir una entrada al RecyclerView.
 3. Añadir la bola que ha salido junto con su color (int) a la BBDD. Esto se realizará mediante la clase PreparedStatement.
 4. Que cada jugador marque en sus cartones el número que ha salido.
 5. Si hay alguien que tenga bingo, se muestre un Toast indicando el/los ganador/es, impidiendo que el juego siga. Esto último, se puede conseguir, simplemente ocultando el display.

NOMBRE	APELLIDOS

Annex I. Obtencion color desde colors.xml

```
getResources().getColor(R.color.black, getContext().getTheme());
```

Annex II. Esquema incompleto de un adaptador para un RecyclerView

```
public class MyRecyclerViewAdapter extends
RecyclerView.Adapter<MyRecyclerViewAdapter.MyViewHolder> {

    private LayoutInflater inflater;
    private Context context;

    public MyRecyclerViewAdapter(@NonNull Context context) {
        this.context = context;
        inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @NonNull
    @Override
    public MyRecyclerViewAdapter.MyViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        View view = inflater.inflate(R.layout.simple_element,parent,false);
        return new MyViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull MyRecyclerViewAdapter.MyViewHolder
holder, int position) {

    }

    @Override
    public int getItemCount() {
        return 0;
    }

    public class MyViewHolder extends RecyclerView.ViewHolder{

        public MyViewHolder(@NonNull View itemView) {
            super(itemView);
        }
    }
}
```

NOMBRE	APELLIDOS

Annex III. Utilization de LinearLayout por codigo

Para diseñar los cartones de los jugadores, podemos realizarlo mediante un LinearLayout. Para ello, la mejor forma es que nuestra clase herede directamente de LinearLayout. De este modo deberemos crear los constructores (el propio IDE nos indicara como hacerlo). Además, dispondremos del siguiente método:

- `setOrientation(int orientation)`: Para definir la orientación. Esta puede ser `LinearLayout.HORIZONTAL` o bien `LinearLayout.VERTICAL`
- `addView(View child)`: metodo para añadir un objeto View. Este se colocará automáticamente, dependiendo de cómo hayamos definido la orientación.

Annex IV. BBDD

Para descargar el conector jdbc recordar añadir las siguientes líneas al pom.xml:

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.23</version>
</dependency>
```

Puedes utilizar el siguiente código para acceder a la base de datos en MariaDB:

```
import com.mysql.cj.jdbc.MySQLDataSource;
import javax.sql.DataSource;

public class MyDataSource {
    public static DataSource getMyDataSource() {
        MySQLDataSource mySQL = new MySQLDataSource();
        mySQL.setURL("jdbc:mysql://localhost:3306/extraordinaria");
        mySQL.setUser("examen");
        mySQL.setPassword("1234");
        return mySQL;
    }
}
```

DDL y DCL de la BBDD:

```
DROP DATABASE IF EXISTS extraordinaria;

CREATE DATABASE extraordinaria;

USE extraordinaria;

CREATE TABLE `bolas` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `numero` int(11) NOT NULL,
```

NOMBRE	APELLIDOS

```
`color` int(11) NOT NULL,  
PRIMARY KEY (`id`),  
UNIQUE KEY `numero` (`numero`)  
);  
  
DELIMITER @@  
  
CREATE PROCEDURE clean()  
BEGIN  
  
    DELETE FROM bolas;  
  
END@@  
  
DELIMITER ;;  
  
DROP USER IF EXISTS examen@'%';  
CREATE USER examen@%' identified by '1234';  
GRANT ALL PRIVILEGES ON extraordinaria.* TO examen@'%';  
GRANT ALL PRIVILEGES ON mysql.proc TO examen@'%';
```