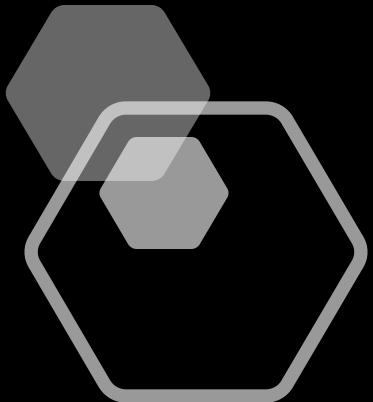




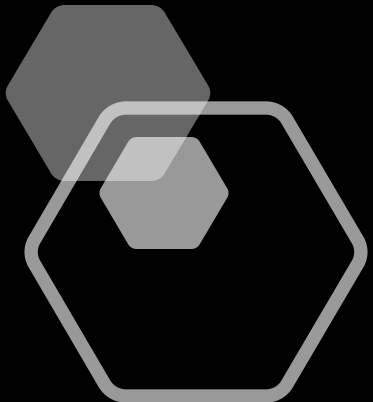
MrJavaBet

- Gabriel Kleiman
- Gabriel Schramm
- Michel Krasner

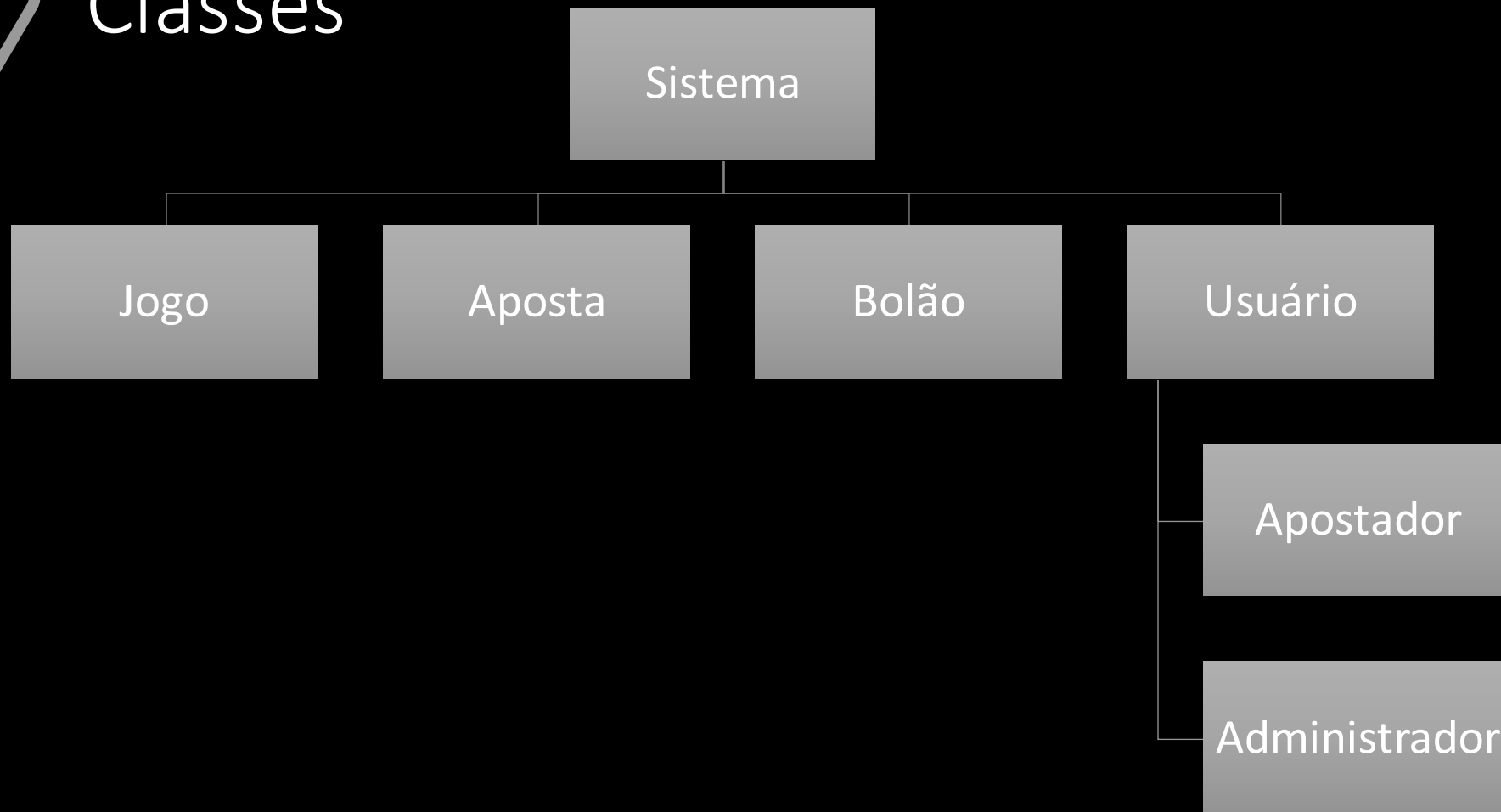


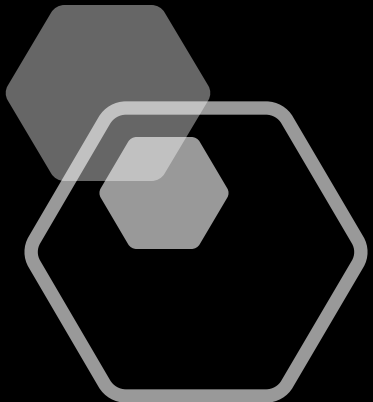
Tarefa: Futebol: Sistema/Jogo de apostas



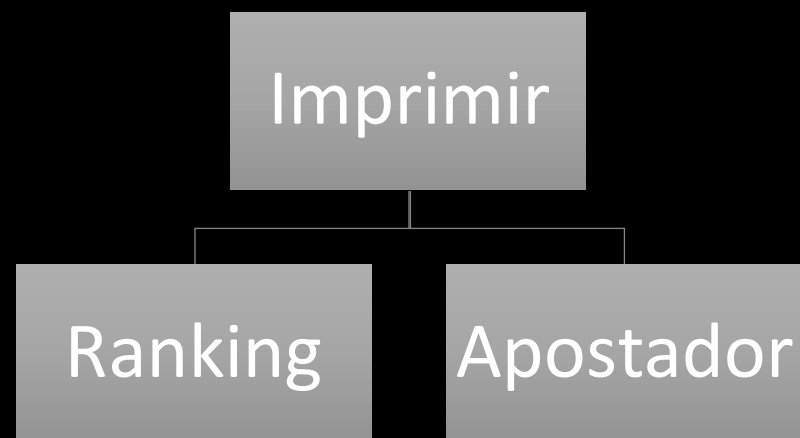
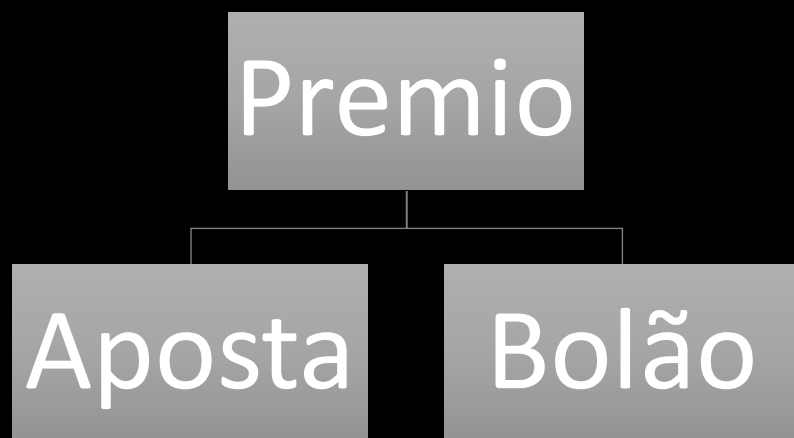


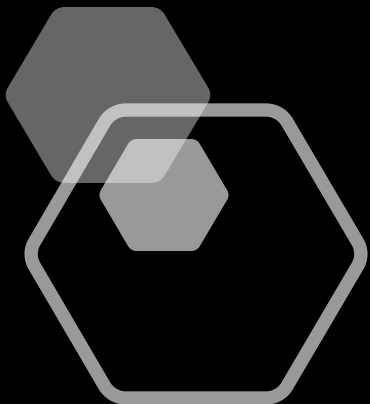
Classes





Interfaces





Aposta

**Brasileirão**
Temporada de 2022

"pais": "Brasil",
"campeonato": "Brasileirão"

...

PARTIDAS

TABELA

ESTATÍSTICAS

< nov.

qua., 9 de nov.

qui., 10 de nov.

sáb., 12 de nov.

dom., 13 de nov.

>

Semana 38

**RB Bragantino**
11-11-16

RB X FC

Fluminense FC
21-7-10 

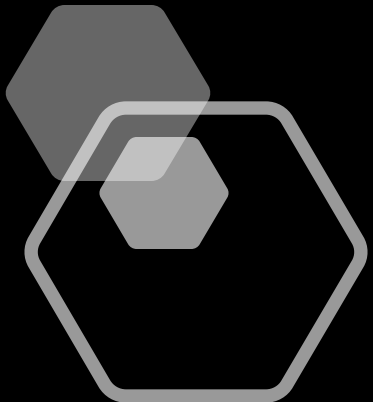
Aposta

Atributos

- jogo
- previsaoPlacar

Métodos

- distribuirPremio
- getResultado



Aposta

```
@Override
public int getResultado() {
    if (this.getJogo().isFinalizado()){
        String timeVencedor = "EMPATE";
        if ((this.getPrevisaoPlacar()[0] == this.getJogo().getPlacar()[0]) &&
            (this.getPrevisaoPlacar()[1] == this.getJogo().getPlacar()[1])){
            return PONTOS_PLACAR_EXATO;
        } else if (this.getJogo().getPlacar()[0] > this.getJogo().getPlacar()[1]) {
            timeVencedor = this.getJogo().getTimes()[0];
        } else if (this.getJogo().getPlacar()[0] < this.getJogo().getPlacar()[1]) {
            timeVencedor = this.getJogo().getTimes()[1];
        }
        if (Objects.equals(this.getPrevisaoTime(), timeVencedor)){
            return PONTOS_TIME;
        }
    }
    return 0;
}
```

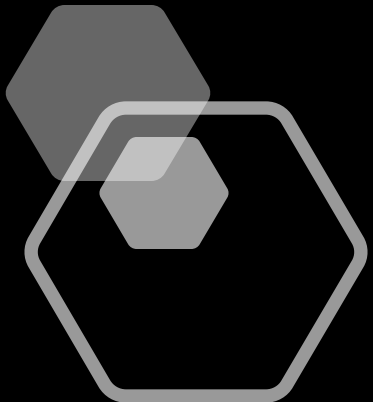
Aposta

Atributos

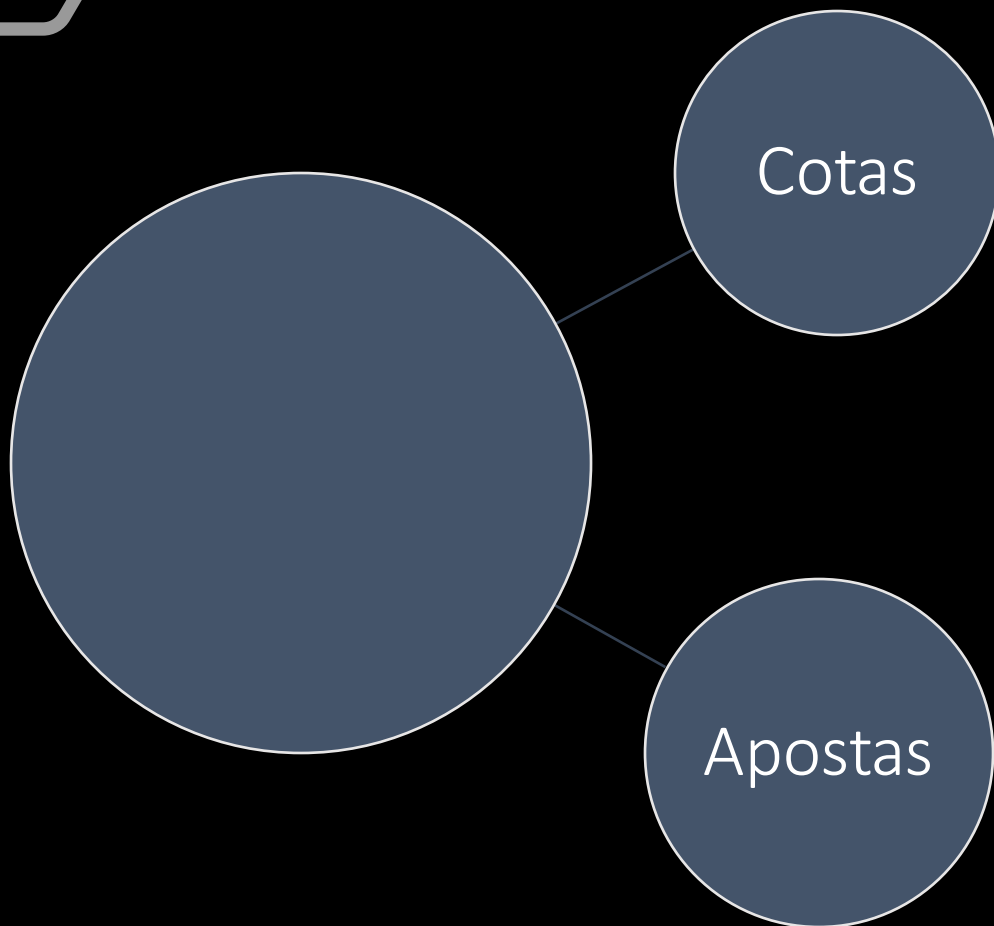
- jogo
- previsaoPlacar

Métodos

- distribuirPremio
- getResultado



Bolão



- Kleiman
- Krasner
- Schramm

- Jogo 1
 - 2 X 1
- Jogo 2
 - 1 x 2
- Jogo 3
 - 0 X 0

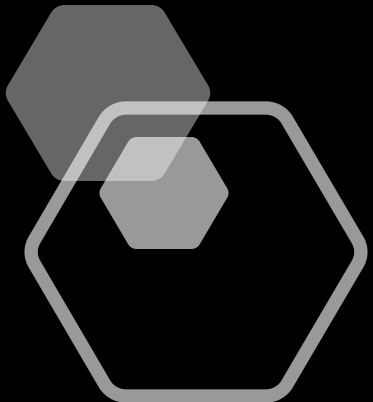
Bolão

Atributos

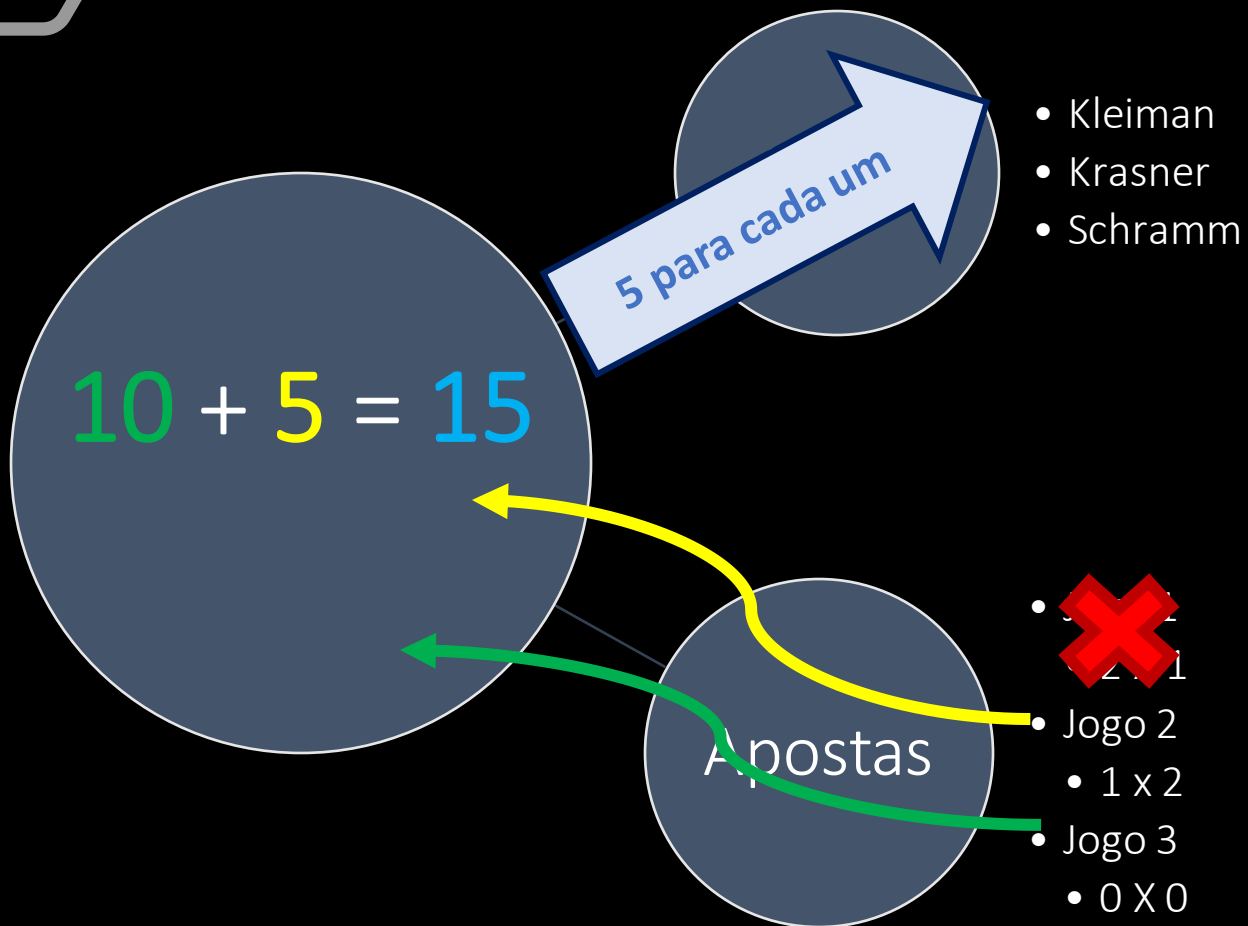
- apostadores
- Jogos
- cotas

Métodos

- distribuirPremio
- getResultado



Bolão



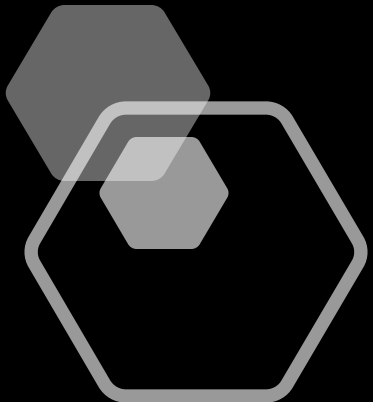
Bolão

Atributos

- apostadores
- Jogos
- cotas

Métodos

- distribuirPremio
- getResultado



Bolão

```
@Override
public boolean distribuirPremio() {
    ArrayList<Boolean> distribuicaoOK = new ArrayList<>();
    for (Aposta aposta : this.getApostas()){
        distribuicaoOK.add(aposta.distribuirPremio( bolao: this));
    }
    if (!distribuicaoOK.contains(false)){
        for (Apostador apostador : this.getApostadores()){
            this.distribuirPremio(apostador);
        }
    }
    return true;
}
```

Bolão

Atributos

- apostadores
- Jogos
- cotas

Métodos

- distribuirPremio
- getResultado

Ranking

```
public class Ranking implements Imprimir {  
  
    2 usages  
    private List<Apostador> apostadores;  
    1 usage  
    private List<Jogo> jogos;  
  
    1 usage  🧑 OmegaTopo  
    public Ranking(List<Jogo> jogos, List<Apostador> apostadores) {  
        this.apostadores = apostadores;  
        this.jogos = jogos;  
    }  
}
```

```
public Ranking(List<Jogo> jogos, List<Apostador> apostadores) {  
    this.apostadores = apostadores;  
    this.jogos = jogos;  
}  
  
1 usage  🧑 OmegaTopo  
public ArrayList<Apostador> getPontuacaoGeral() {  
    return apostadores.stream()  
        .sorted(Comparator.reverseOrder())  
        .collect(Collectors.toCollection(ArrayList::new));  
}  
  
1 usage  🧑 OmegaTopo  
@Override  
public void imprimir() {  
    int i = 1;  
    for (Apostador apostador : this.getPontuacaoGeral()) {  
        System.out.println("\t\t" + i + " " + apostador.getNome() + " -- \t" + apostador.getPontos() + " Pontos ");  
        i++;  
    }  
}  
}
```

Jogo

```
public class Jogo {  
    1 usage  
    private static Scanner scanner = new Scanner(System.in);  
    4 usages  
    private String campeonato;  
    4 usages  
    private String pais;  
    6 usages  
    private String[] times = {"", ""};  
    3 usages  
    private int[] placar = {0, 0};  
    3 usages  
    private boolean finalizado;  
  
    5 usages  ⬆ OmegaTopo +1  
    public Jogo(String campeonato, String pais, String time1, String time2) {  
        times[0] = time1;  
        times[1] = time2;  
        this.campeonato = campeonato;  
        this.pais = pais;  
        this.finalizado = false;  
    }  
}
```

```
public void setPais(String pais) { this.pais = pais; }  
  
22 usages  ⬆ Gabriel Kleiman  
public String[] getTimes() { return times; }  
  
2 usages  ⬆ OmegaTopo +1  
public void setTimes(int index, String nomeTime) { this.times[index] = nomeTime; }  
  
12 usages  ⬆ Gabriel Kleiman  
public int[] getPlacar() { return placar; }  
  
9 usages  ⬆ OmegaTopo +1  
public void setPlacar(int placar1, int placar2) {  
    this.placar[0] = placar1;  
    this.placar[1] = placar2;  
}  
  
7 usages  ⬆ Gabriel Kleiman  
public boolean isFinalizado() { return finalizado; }  
  
2 usages  ⬆ Gabriel Kleiman  
public void setFinalizado(boolean finalizado) { this.finalizado = finalizado; }
```

Administrador

1 usage · Gabriel Kleiman +1

```
public void menuAdministrador() {
    System.out.println("\n\tBEM VINDO " + usuarioAtivo.getEmail().toUpperCase());
    String opcao = "";
    while (!opcao.equals("0")) {
        System.out.println("\n\tMENU ADMINISTRADOR\n-----");
        System.out.println("\t1 - CRUD Apostador");
        System.out.println("\t2 - CRUD Administrador");
        System.out.println("\t3 - CRUD Aposta");
        System.out.println("\t4 - CRUD Bolão");
        System.out.println("\t5 - CRUD Jogo");
        System.out.println("\t6 - GERAR APOSTADORES TESTE");
        System.out.println("\t7 - GERAR JOGOS TESTE");
        System.out.println("\t8 - GERAR BOLÃO TESTE");
        System.out.println("\t0 - Sair");
        System.out.print("\tEscolha uma opção: ");
        opcao = scanner.nextLine();
        switch (opcao) {
            case "1":
                crudApostador();
```

Apostador

```
1 usage  Gabriel Kleiman +1
public void menuApostador() {
    System.out.println("\n\tBEM VINDO " + usuarioAtivo.getEmail().toUpperCase() + "!\n---");
    String opcao = "";
    while (!opcao.equals("0")) {
        System.out.println("\n\tMENU APOSTADOR\n-----");
        System.out.println("\t1 - Realizar uma aposta");
        System.out.println("\t2 - Comprar cota de bolão");
        System.out.println("\t3 - Verificar resultado de apostas");
        System.out.println("\t4 - Verificar resultado de bolões");
        System.out.println("\t5 - Consultar meus dados");
        System.out.println("\t6 - Depositar");
        System.out.println("\t7 - Trocar por prêmio");
        System.out.println("\t0 - Sair");
        System.out.print("\tEscolha uma opção: ");
        opcao = scanner.nextLine();
        switch (opcao) {
            case "1":
                System.out.println("\nRealizar uma aposta");
                apostadorRealizarAposta();
            
```

Bolão

```
@Override
public int getResultado() {
    return this.getApostas().stream() Stream<Aposta>
        .mapToInt(Aposta::getResultado) IntStream
        .sum();
}
```

```
@Override
public boolean distribuirPremio(Apostador apostador) {
    int pontoPorCota = this.getPontos()/this.getCotas();
    this.setFinalizado(true);
    apostador.setPontos(apostador.getPontos() + pontoPorCota);
    return true;
}
```

```
@Override
public boolean distribuirPremio() {
    ArrayList<Boolean> distribuicaoOK = new ArrayList<>();
    for (Aposta aposta : this.getApostas()){
        distribuicaoOK.add(aposta.distribuirPremio( bolao: this));
    }
    if (!distribuicaoOK.contains(false)){//Contem falso
        for (Apostador apostador : this.getApostadores()){
            this.distribuirPremio(apostador);
        }
    }
    return true;
}
```