**Classe: Cadastro**

**Atributos:**

Não possui.

**Métodos:**

**capturaDados**

**Parâmetro**: lista: ArrayList <Anuncio>.

**Return**: Anuncio.

Esse método tem o objetivo de capturar e armazenar o nome do anuncio, cliente e o valor diário investido. Após a verificação da duplicidade de nomes de anuncio, o método chama outros dois métodos para a capturar e armazenar a data de início e término do anuncio. Ao final, ele cria e retorna um objeto Anuncio.

**dataInicial**

**Parâmetro**: Não possui.

**Return**: String

Esse método tem o objetivo de capturar, armazenar e verificar se é válida a data inicial do anuncio. Ela também formata a data armazenada para “dd/mm/aaaa” quando necessário.

**dataFinal**

**Parâmetro**: dataInicial: String.

**Return**: String

Esse método tem o objetivo de capturar, armazenar e verificar se é válida a data final do anuncio. Ela também formata a data armazenada para “dd/mm/aaaa” quando necessário e compara as datas final e inicial do anuncio

**pesquisa**

**Parâmetro**: lista: ArrayList <Anuncio>, aux: boolean

**Return**: ArrayList <Anuncio>

Esse método tem o objetivo de fazer uma filtragem de anúncios através dos seguintes parâmetros:

1. Nome do anuncio
2. Nome do cliente
3. Intervalo de data

Ele recebe como parâmetro a lista de Anuncios adicionados e uma variável auxiliadora, pois esse método possui dois comportamentos diferentes dependendo de onde é invocado.

A primeira forma de invoca-lo é através do Menu Principal na opção “2 – Procurar por um anuncio”. Nesse caso, a variável auxiliar recebe valor **false** e o usuário pode escolher entre pesquisar pelo (1) nome do anuncio, (2) nome do cliente ou (3) intervalo de data.

A segunda forma de invoca-lo é através da opção do Menu Principal “4 – Calcular alcance por data”. Nesse caso, a variável auxiliar recebe valor **true** e o usuário é obrigado a digitar o nome do anuncio para fazer a pesquisa.

Essa solução foi implementada pelo motivo de que no primeiro caso o usuário pode obter como retorno vários anúncios (por isso retorno em ArrayList) e no segundo caso o programa limita o usuário a escolher apenas um anuncio por vez para fazer o cálculo de alcance por data, já que não é permitido duplicidade no nome do anuncio.

**calculaDias**

**Parâmetro**: dataInicial: String, dataFinal: String.

**Return**: int.

Esse método tem o objetivo de retornar à quantidade de dias existentes entre a data inicial e a data final. O método divide esse cálculo entre:

1. Diferença entre os dias do mês;
2. Diferença entre os meses;
3. Diferença entre os anos.

Ao final, o programa irá operá-los de acordo com a situação que cada um se encaixa.

**anuncioEncontrado**

**Parâmetro**: find: Anuncio.

**Return**: int.

Esse método tem o objetivo de calcular a quantidade de dias entre duas datas filtradas através da opção “4 – Calcular alcance por data” do Menu Principal.

Para ser invocado esse método precisará primeiro passar pelo método **pesquisa** descrito anteriormente. O resultado obtido na chamada do método **pesquisa** será passado como parâmetro para este, que por sua vez irá verificar se as datas informadas como filtro pelo usuário correspondem a um período em que o anuncio está ativo. Em caso positivo, o método **calculaDias** é acionado e o cálculo é feito.

**dataValida**

**Parâmetro**: data: String.

**Return**: int.

Esse método tem o objetivo de verificar se uma data digitada é válida. Para isso ele utiliza try catch para tentar uma conversão de String para LocalDate, se a conversão for bem sucedida ele retorna **true**, se for mal sucedida retorna **false**. Esse método serve de auxiliar para todos os métodos que trabalham com data.