



POO - Programação Orientada a Objetos  
TADS - Análise Desenvolvimento de Sistemas

## **Implementar o Encapsulamento**

Professor: Carlos Veríssimo  
Componentes: Iago Vinycius e Vitoria Cruz

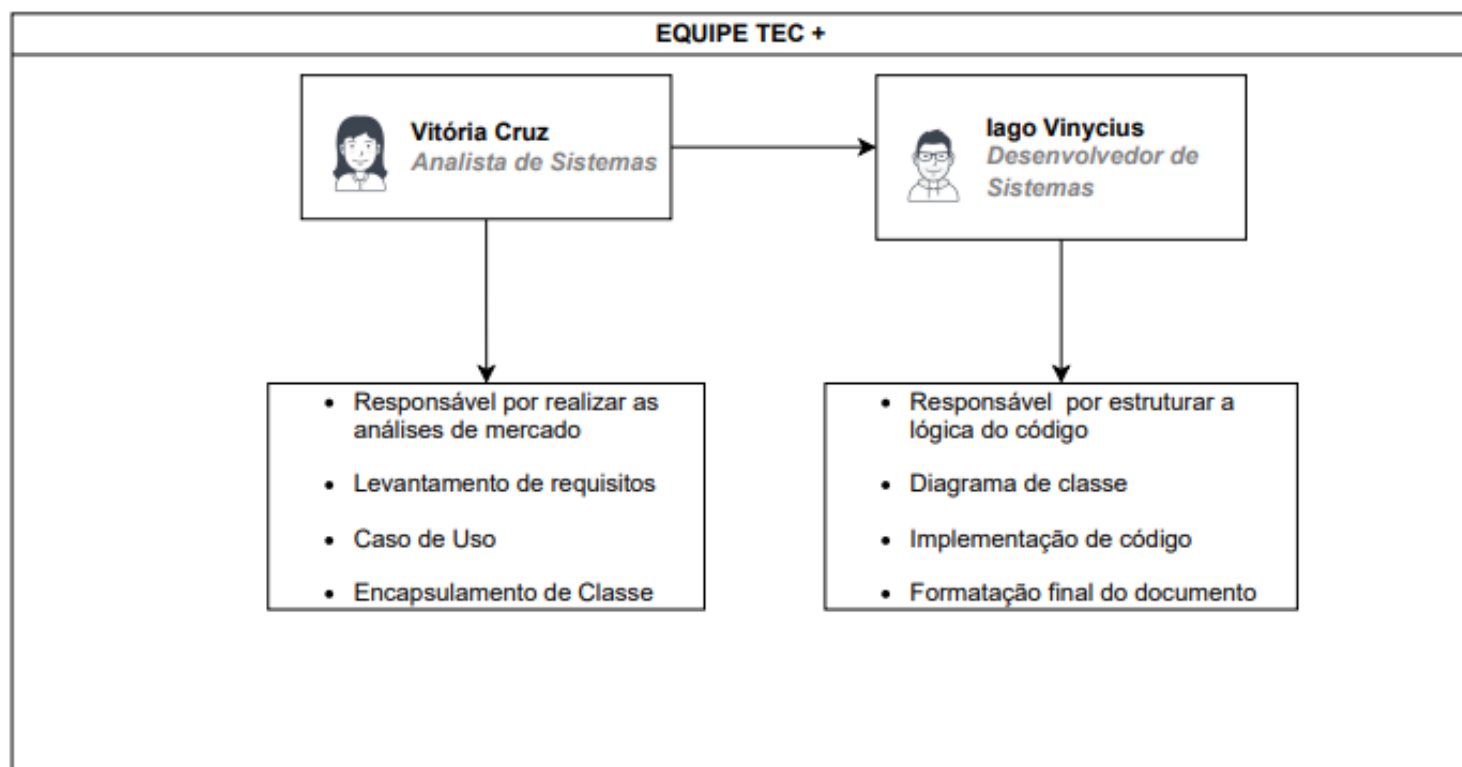
**São Paulo**

2023

## Índice

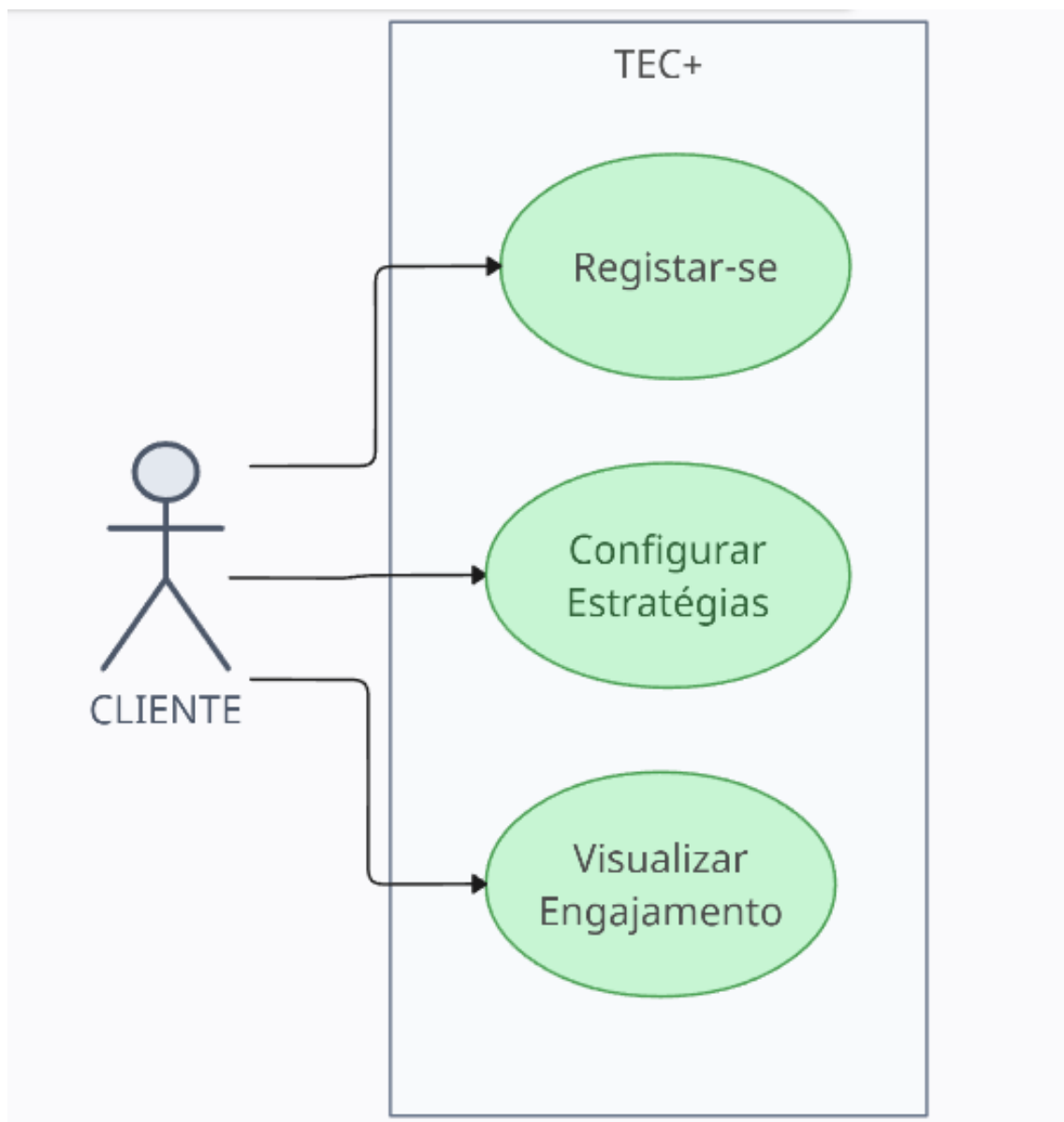
<b>Componentes da Equipe .....</b>	<b>1</b>
<b>Especificações do Sistema .....</b>	<b>Erro! Indicador não definido.</b>
<b>1- Cenário .....</b>	<b>Erro! Indicador não definido.</b>
<b>2- Planejamento .....</b>	<b>Erro! Indicador não definido.</b>
<b>3- Situação do Problema .....</b>	<b>Erro! Indicador não definido.</b>
<b>4- Solução de Software .....</b>	<b>Erro! Indicador não definido.</b>
4.1 - Requisitos do Cliente .....	<b>Erro! Indicador não definido.</b>
4.2 - Requisitos Não Funcionais .....	<b>Erro! Indicador não definido.</b>
<b>5 - Casos de Uso (UC) .....</b>	<b>2</b>
5.1 - Diagrama de Caso de Uso - Visão Geral .....	2
5.2 - Caso de Uso - Caso de Uso #1 - Registrar-se .....	3
5.3 - Caso de Uso - Caso de Uso #2 - Configurar Estratégias .....	5
5.4 - Caso de Uso - Caso de Uso #3 - Visualizar Engajamento .....	8
<b>6- Diagrama de Classes .....</b>	<b>10</b>
<b>Conclusão .....</b>	<b>Erro! Indicador não definido.</b>

## Componentes da Equipe



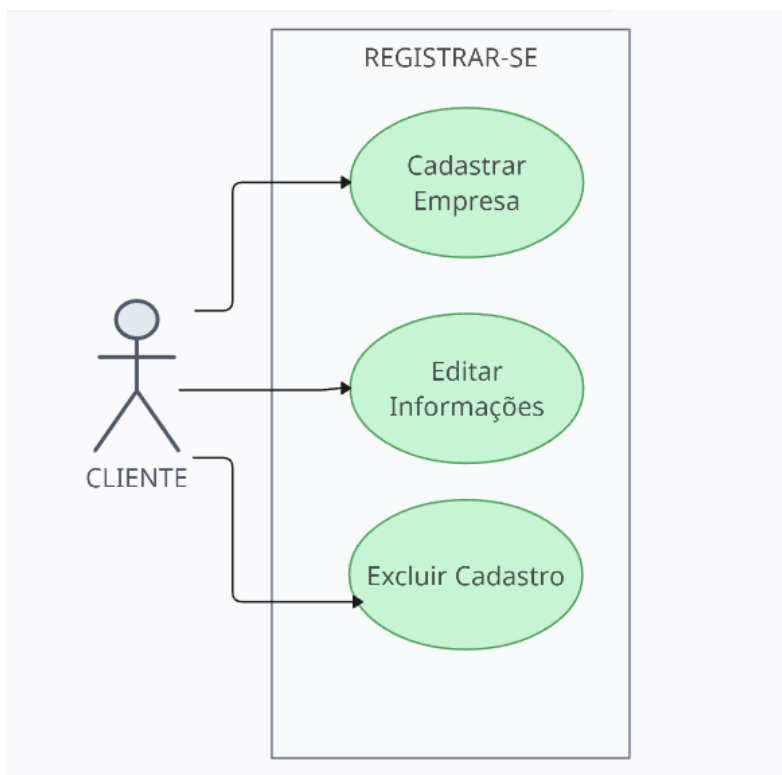
## 1 - Casos de Uso (UC)

### 1.1 - Diagrama de Caso de Uso - Visão Geral



## 1.2 - Caso de Uso - Caso de Uso #1 - Registrar-se

### 1.2.1 - Diagrama do Caso de Uso #1



### 1.2.2 - Detalhamento do Caso de Uso #1.1 - Cadastro

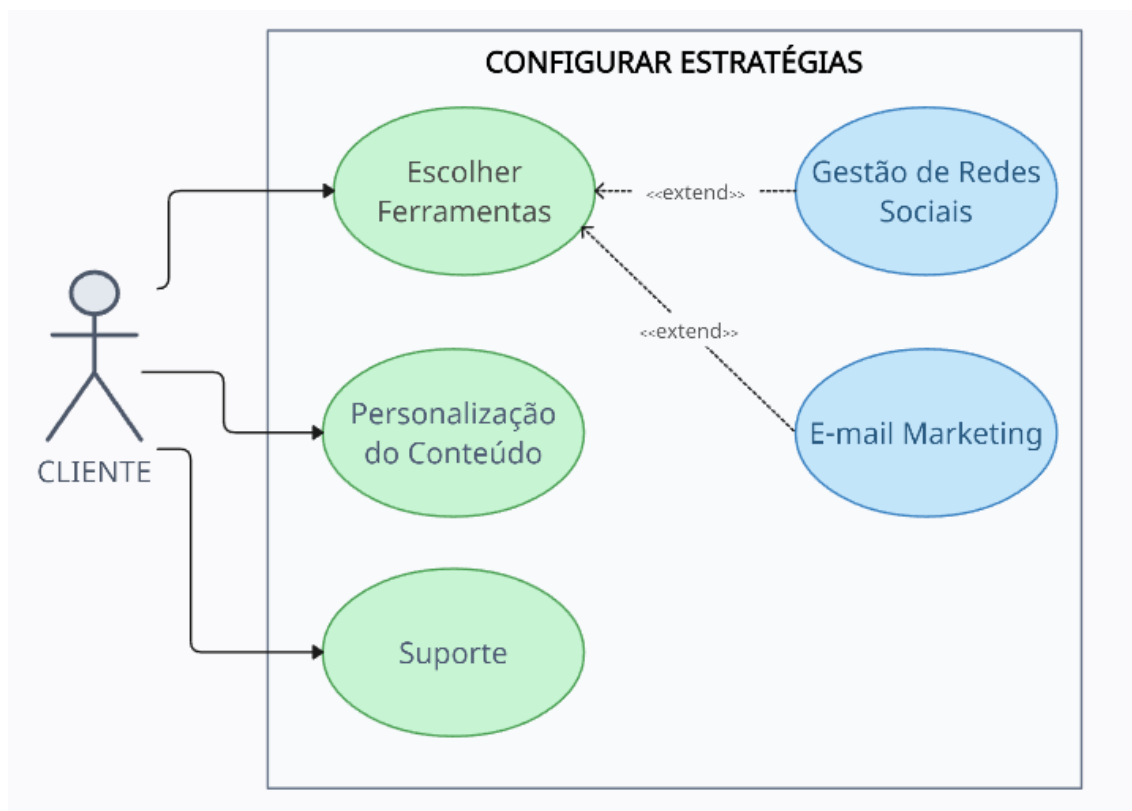
<b>Nome do Caso de Uso:</b>	Registrar-se
<b>Atores:</b>	Cliente
<b>Tigger:</b>	Necessidade do cliente em utilizar nossos serviços.
<b>Pré-requisito:</b>	Ser uma empresa de tecnologia
<b>Fluxo de Eventos:</b>	Cliente acessa nosso site, Clica no botão de “registre-se” e preenche os campos obrigatórios.

**1.2.3 - Detalhamento do Caso de Uso #1.1 - Editar Cadastro**

<b>Nome do Caso de Uso:</b>	Editar Cadastro
<b>Atores:</b>	Cliente
<b>Tigger:</b>	Conferir os dados informados
<b>Pré-requisito:</b>	Ter dados inseridos
<b>Fluxo de Eventos:</b>	Cliente acessa nosso site, clica no botão de “registre-se”, preenche os campos obrigatórios, erro de digitação ou esquecimento de campo obrigatório, edição ou adição dos dados.

### 1.3 - Caso de Uso - Caso de Uso #2 - Configurar Estratégias

#### 1.3.1 - Diagrama do Caso de Uso #2



#### 1.3.2 - Detalhamento do Caso de Uso #2.1 - Configurar estratégias

<b>Nome do Caso de Uso:</b>	Escolher Ferramentas
<b>Atores:</b>	Cliente
<b>Tiger:</b>	Escolher os métodos de expansão da marca que mais se adeque com a empresa
<b>Pré-requisito:</b>	Ter realizado o cadastro
<b>Fluxo de Eventos:</b>	Cliente realiza login no site, Entra na página “configurar estratégias”, clica em “escolher ferramentas” e seleciona as que mais agradam.

### **1.3.3 - Detalhamento do Caso de Uso #2.2 - Configurar estratégias**

<b>Nome do Caso de Uso:</b>	Gestão de Redes sociais
<b>Atores:</b>	Cliente
<b>Tiger:</b>	A identificação da empresa com essa forma de marketing
<b>Pré-requisito:</b>	Estar logado na plataforma
<b>Fluxo de Eventos:</b>	Cliente realiza login no site, Entra na página “configurar estratégias”, clica em “escolher ferramentas” e seleciona a opção de “gestão de redes sociais”.

### **1.3.4 - Detalhamento do Caso de Uso #2.3 - Configurar estratégias**

<b>Nome do Caso de Uso:</b>	E-mail marketing
<b>Atores:</b>	Cliente
<b>Tiger:</b>	A identificação da empresa com essa forma de marketing
<b>Pré-requisito:</b>	Estar logado na plataforma
<b>Fluxo de Eventos:</b>	Cliente realiza login no site, Entra na página “configurar estratégias”, clica em “escolher ferramentas” e seleciona a opção de “e-mail marketing”.



### **1.3.5 - Detalhamento do Caso de Uso #2.4 - Configurar estratégias**

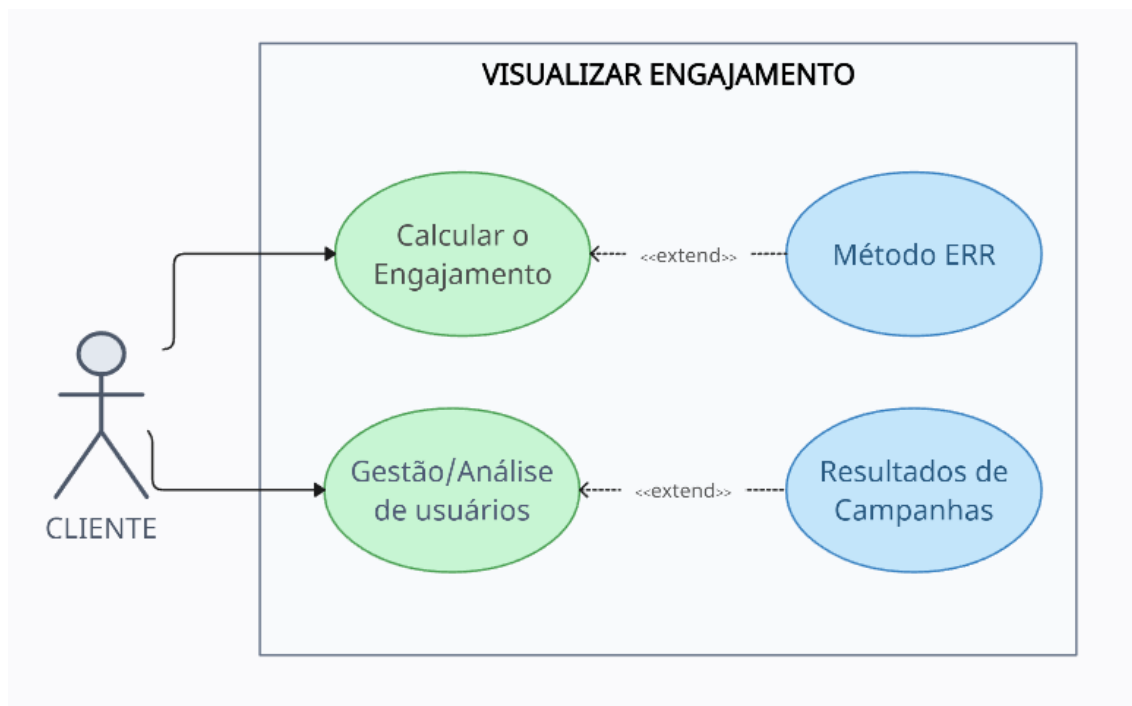
<b>Nome do Caso de Uso:</b>	Personalização de conteúdo
<b>Atores:</b>	Cliente
<b>Tiger:</b>	Pré-definir o conteúdo dos anúncios
<b>Pré-requisito:</b>	Ter escolhido no mínimo uma opção de expansão de marca
<b>Fluxo de Eventos:</b>	Cliente realiza login no site, Entra na página “configurar estratégias”, clica em “personalização de conteúdo” e responde as questões pré-definidas para a criação de um modelo de anúncio.

### **1.3.6 - Detalhamento do Caso de Uso #2.5 - Configurar estratégias**

<b>Nome do Caso de Uso:</b>	Suporte
<b>Atores:</b>	Cliente
<b>Tiger:</b>	Ter encontrado alguma dificuldade na navegação da plataforma
<b>Pré-requisito:</b>	Ter realizado o cadastro
<b>Fluxo de Eventos:</b>	Cliente loga no site, clica em “suporte”, seleciona o processo que está tendo dificuldade.

## 1.4 - Caso de Uso - Caso de Uso #3 - Visualizar Engajamento

### 1.4.1 - Diagrama do Caso de Uso #3



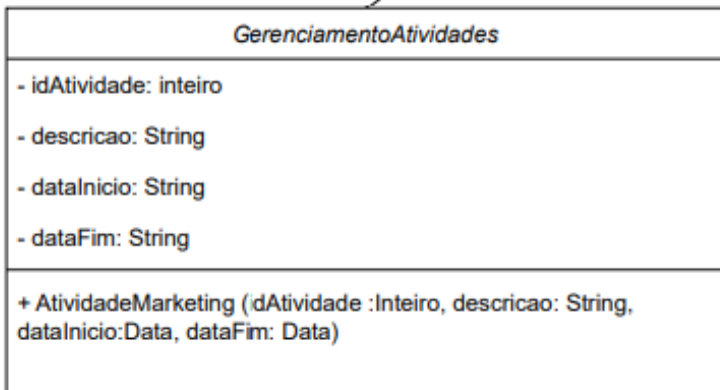
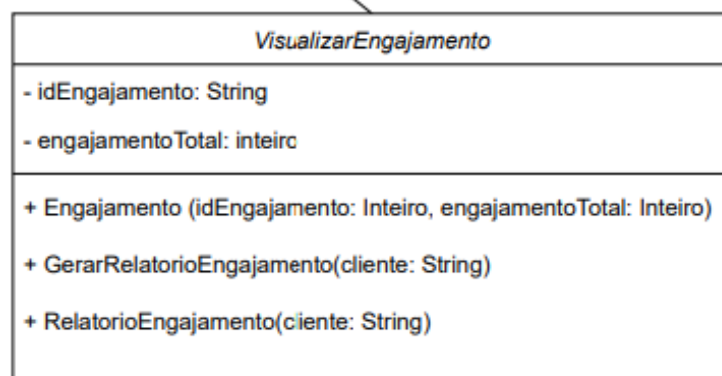
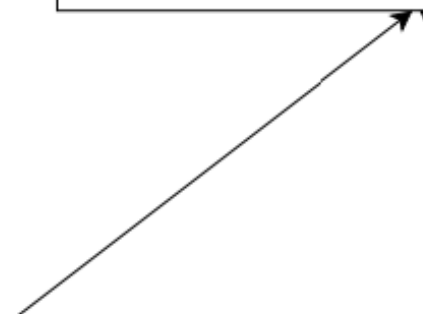
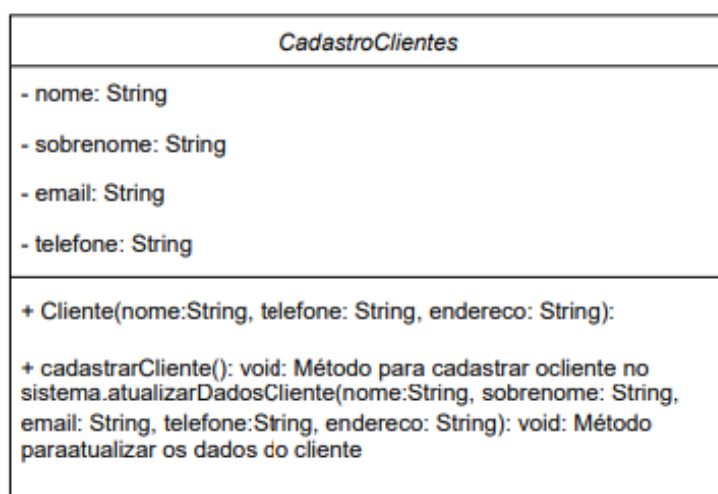
### 1.4.2 - Detalhamento do Caso de Uso #3.1 - Calcular Engajamento

<b>Nome do Caso de Uso:</b>	Calcular Engajamento
<b>Atores:</b>	Cliente
<b>Tigger:</b>	Calcular
<b>Pré-requisito:</b>	Obter resultados através da utilização do sistema
<b>Fluxo de Eventos:</b>	Cliente consegue medir as interações por alcance de uma postagem utilizando o método <b>ERR = soma de todas as interações/alcance do post.</b>

**1.4.3 - Detalhamento do Caso de Uso #3.2 - Gestão/Análise de Usuários**

<b>Nome do Caso de Uso:</b>	Gestão/Análise de Usuários
<b>Atores:</b>	Cliente
<b>Tigger:</b>	Analisar Clientes
<b>Pré-requisito:</b>	Possuir mais que 10 compradores
<b>Fluxo de Eventos:</b>	Cliente consegue acesso analisar o comportamento dos seus interessados

## 2- Diagrama de Classes



### 3 - Implantação do Encapsulamento

#### Classe: GerenciarAtividades

```
package N1;
public class GerenciarAtividades {

    private int idAtividade; // Atributo Privado
    private String descricao; // Atributo Privado
    private String dataInicio; // Atributo Privado
    private String dataFim; // Atributo Privado

    public int getIdAtividade() { // Método público de acesso (getter)
para idAtividade
        return idAtividade;
    }

    public void setIdAtividade(int idAtividade) { // Método público de
acesso (setter) para idAtividade
        this.idAtividade = idAtividade;
    }

    public String getDescricao() { // Método público de acesso
(getter) para descricao
        return descricao;
    }

    public void setDescricao(String descricao) { // Método público de
acesso (setter) para descricao
        this.descricao = descricao;
    }

    public String getDataInicio() { // Método público de acesso
(getter) para dataInicio

        return dataInicio;
    }

    public void setDataInicio(String dataInicio) { // Método público
de acesso (setter) para dataInicio
        this.dataInicio = dataInicio;
    }

    public String getDataFim() { // Método público de acesso (getter)
para dataFim
        return dataFim;
    }

    public void setDataFim(String dataFim) { // Método público de
acesso (setter) para DataFim
        this.dataFim = dataFim;
    }

    public GerenciarAtividades() {} // Metodo Construtor

    public GerenciarAtividades(int idAtividade, String descricao,
String dataInicio, String dataFim) { // Metodo Construtor com
parametros
        this.idAtividade = idAtividade;
        this.descricao = descricao;
        this.dataInicio = dataInicio;
        this.dataFim = dataFim;
    }
}
```

## Classe: Cliente

```
package N1;

public class Cliente {

    private int ID;
    private String Nome; // Atributo Privado
    private String Senha; // Atributo Privado
    private String Email; // Atributo Privado
    private String Telefone; // Atributo Privado
    private String NomeEmpresa; // Atributo Privado

    public Cliente() {}

    public int getID() { // Método público de acesso (getter) para ID
        return ID;
    }

    public void setID(int ID) { // Método público de acesso (setter)
para ID
        this.ID = ID;
    }

    public Cliente(int id, String nome, String nomeEmpresa, String
email, String senha, String telefone) {
        this.ID = id;
        this.Nome = nome;
        this.NomeEmpresa = nomeEmpresa;
        this.Email = email;
        this.Senha = senha;
        this.Telefone = telefone;
    }

    public String getNome() { // Método público de acesso (getter)
para Nome
        return Nome;
    }

    public void setNome(String nome) { // Método público de acesso
(setter) para Nome
        this.Nome = nome;
    }

    public String getNomeEmpresa() { // Método público de acesso
(getter) para NomeEmpresa
        return NomeEmpresa;
    }

    public void setNomeEmpresa(String nomeEmpresa) { // Método público
de acesso (setter) para NomeEmpresa
        NomeEmpresa = nomeEmpresa;
    }

    public String getEmail() { // Método público de acesso (getter)
para E-mail
        return Email;
    }
}
```

```

        public void setEmail(String email) {
// Método público de acesso (setter) para E-mail
            Email = email;
        }

        public String getTelefone() { // Método público de acesso (getter)
para Telefone
            return Telefone;
        }

        public void setTelefone(String telefone) { // Método público de
acesso (setter) para Telefone
            Telefone = telefone;
        }

        public String getSenha() { // Método público de acesso (getter)
para Senha
            return Senha;
        }

        public void setSenha(String senha) { // Método público de acesso
(setter) para Senha
            Senha = senha;
        }

        @Override
        public String toString() {
            return "ID: " + ID + " | Nome: " + Nome + " | Nome da Empresa:
" + NomeEmpresa + " | E-mail: " + Email + " | Telefone: " + Telefone;
        }
    }

```

## Classe: VisualizarEngajamento

```
public class VisualizarEngajamento {  
  
    private int idEngajamento; // Atributo Privado  
    private int engajamentoTotal; // Atributo Privado  
  
    public int getIdEngajamento() { // Método público de acesso  
        (getter) para idEngajamento  
        return idEngajamento;  
    }  
  
    public void setIdEngajamento(int idEngajamento) { // Método  
        público de acesso (setter) para idEngajamento  
        this.idEngajamento = idEngajamento;  
    }  
  
    public int getEngajamentoTotal() { // Método público de acesso  
        (getter) para engajamentoTotal  
  
        return engajamentoTotal;  
    }  
  
    public void setEngajamentoTotal(int engajamentoTotal) { // Método  
        público de acesso (setter) para engajamentoTotal  
        this.engajamentoTotal = engajamentoTotal;  
    }  
  
    public VisualizarEngajamento () {}  
  
    public static void Engajmaento (int idEngajamento, int  
    engajamentoTotal) {  
  
    }  
  
    public static void GerarRelatorioEngajamento (String cliente) {  
  
    }  
  
    public static void RelatorioEngajamento (String cliente) {  
  
    }  
}
```