

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

ESCOLA DE ENGENHARIA

DEPARTAMENTO DE ELETRÔNICA E DE COMPUTAÇÃO

**SISTEMA DE GERENCIAMENTO DE BANNER  
PARA SOFTWARE DESKTOP**

**Autor:**

---

Rafael Toshiro Sugii

**Orientador:**

---

Ph. D. Sergio Barbosa Villas-Boas

**Co-orientador:**

---

Ph. D.

**Examinador:**

---

Ph. D.

**DEL**  
**Março de 2008**

## **DEDICATÓRIA**

Dedico este trabalho à minha família por valorizar o meu futuro.

## **AGRADECIMENTO**

Primeiramente, gostaria de agradecer aos meus pais pelo apoio e paciência que depositaram em mim ao longo dos muitos anos de estudo até a universidade.

Aos meus “Amigos do Fundão” que sempre me motivaram durante o curso. Ao meu orientador de projeto de fim de curso Sergio B. Villas-Boas, pela paciência e apoio ao meu trabalho.

Ao meu orientador acadêmico Luiz Wagner Pereira Biscainho por não deixar de lado meus constantes pedidos de alterações em disciplinas. E à todos os meus amigos que sempre me apoiaram durante a vida que sem eles, não conseguiria terminar o curso.

Ao meu orientador de iniciação científica Geraldo Antônio Guerrera Cidade pelo apoio na minha formação e por toda a paciência dedicada ao meu trabalho.

E por fim, ao povo brasileiro por possibilitar o meu estudo em uma universidade pública como a UFRJ.

## **RESUMO**

Este trabalho tem por objetivo a criação de um modelo de negócio baseado no desenvolvimento de um sistema gerenciador de banner.

O sistema de banner irá permitir a gerência e o acompanhamento de propagandas que serão utilizadas na forma de banner. O programa irá gerar relatórios de acordo com a necessidade do usuário. O sistema será gerenciado através de um navegador de internet.

Para a confecção do sistema e para melhor entendimento dos mesmo, serão utilizados modelos e conceitos baseado na engenharia de software.

## **PALAVRAS-CHAVE**

Gerenciador de banner

wxWidgets

Kepler project

C++

## SUMÁRIO

## ÍNDICE DE FIGURAS

# **1INTRODUÇÃO**

## **1.1 MOTIVAÇÃO**

Dentro do mercado de software, a viabilização do modelo de negócios possui um problema de distribuição. Uma das abordagens do modelo de negócios para uma empresa de software, é o modelo de adware.

Nesse modelo, o cliente final usa um sistema 100% gratuito mas ele visualiza uma propaganda enquanto utiliza o software. A software house ganha dinheiro através da venda do espaço publicitário nos banners que o usuário final visualiza e a empresa de software precisa trabalhar com uma agência de propaganda que fatura a partir dos clientes interessados em uma campanha publicitária.

Esses clientes vão gerar demandas que precisam ser gerenciadas. Os problemas de gerencia para esta finalidade, motivaram a criação desse trabalho, que visa solucionar este problema através de uma aplicação genérica.

Este trabalho foca os programas GUI (com interface com o usuário em computadores desktop).

## **1.2 DESCRIÇÃO DO PROJETO**

O objetivo deste projeto é disponibilizar uma ferramenta capaz de gerar relatórios de acesso e uso de propagandas e através dele, criar um modelo de negócio capaz simples e funcional.

O sistema irá contar com um administrador para cadastro de clientes onde cada cliente terá uma relação de mídias vinculados aos mesmos.

O cliente irá receber um acesso onde ele poderá acompanhar os relatórios de acesso e uso e dessa forma, acompanhar a evolução de acesso do seu produto.

Este sistema será disponibilizado através da internet e o usuário poderá acessar o sistema apenas utilizando um navegador. A interface deverá ser intuitiva e para tanto, será utilizado padrões e definições de usabilidade e codificação definidos pelo consórcio de empresas de tecnologia conhecido como World Wide Web Consortium- W3C.



## 2 FUNDAMENTOS TEÓRICOS

### 2.1 PROPAGANDA OU PUBLICIDADE

As técnicas de propaganda foram cientificamente organizadas e aplicadas primeiramente pelo jornalista Walter Lippman e pelo psicólogo Edward Bernays (sobrinho de Sigmund Freud, no início do século XX). Durante a Primeira Guerra Mundial, Lippman e Bernays foram contratados pelo presidente dos Estados Unidos Woodrow Wilson para influenciar a opinião pública para entrar na guerra ao lado da Inglaterra.

A atual indústria das Relações Públicas é uma derivação direta do trabalho de Lippman e Bernays e continua a ser usada largamente pelo governo dos Estados Unidos.

A Segunda Guerra Mundial viu o uso contínuo da propaganda como arma de guerra, tanto pelo propagandista de Hitler Joseph Goebbels como pelo Comitê de Guerra Político-Executivo inglês.

A maioria da propaganda na Alemanha foi produzida pelo Ministério da Conscientização Pública e Propaganda ("Promi" na abreviação alemã).

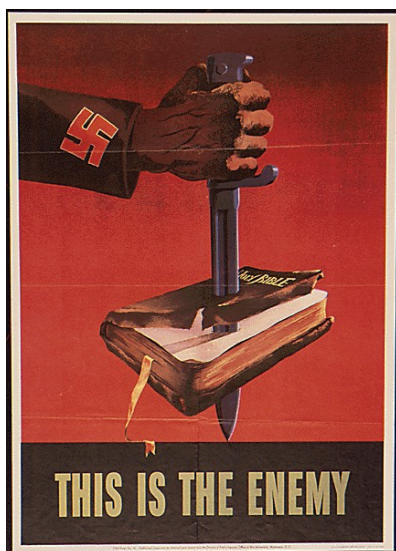


Figura 1: Propaganda nazista

Os nazistas acreditavam na propaganda como uma ferramenta vital para o atingimento de seus objetivos. Adolf Hitler, o Führer da Alemanha, ficou impressionado com o poder da propaganda Aliada durante a Primeira Guerra Mundial e acreditava ter ela sido a causa principal do colapso moral e das revoltas no front alemão e na Marinha em 1918.

De lá para cá os meios de criação de publicidade evoluíram bastante mas sua idéia continuou baseada na exploração do seu significado, do latim moderno, propaganda quer dizer "para ser espalhado".

Nos tempos mais atuais, com a evolução da internet, surgiu mais um meio de comunicação com alcance em massa e internacional. Viu-se a oportunidade de espalhar a divulgação através das propagandas e foi conhecida como banner.

O primeiro banner da história foi criado pela Hotwired para a empresa de telecomunicações norte-americana AT&T. Entrou no ar em 25 de outubro de 1994.



Figura 2: Primeiro banner

O sucesso foi instantâneo, mas as dúvidas sobre a viabilidade da Internet como uma mídia publicitária perduraram por muito tempo.

A Hotwired foi a primeira empresa em vender publicidade na web. Foi ela quem criou o termo "banner", até hoje o modelo publicitário de Internet mais utilizado em todo o mundo. E também foi a primeira a utilizar o sistema de remuneração por cliques, em substituição à remuneração por visualizações. Os responsáveis pela criação da Hotwired foram Andrew Anker, primeiro CEO da empresa, e Rich Boyce, um publicitário visionário que abriu as portas para a propaganda na Internet. A empresa fazia parte da Wired Ventures, que publica a revista Wired, e foi vendida em 1999 para a Lycos.

## 2.2 PLATAFORMA DE DESENVOLVIMENTO WEB – KEPLER

O sistema de gerenciamento de mídias utiliza uma plataforma de desenvolvimento para rodar. Kepler é uma plataforma de código aberto para o desenvolvimento de aplicações Web. É implementado como um conjunto de componentes da linguagem de programação Lua e oferece as mesmas vantagens que Lua: é simples, portátil, leve and extensível.

Por ser independente de sistema operacional, Kepler pode ser utilizado em Windows, Linux, OSX e diversos outros sistemas operacionais. Outro benefício que Kepler herdou de Lua é uma implementação extremamente eficiente. Benchmarks independentes como o Gentoo: Intel® Pentium® 4 Computer Language Shootout avaliam Lua como mais rápida do que outras plataformas de desenvolvimento como Python, Perl, PHP e Ruby.

A Plataforma Kepler foi desenvolvido através do Projeto Kepler, que é coordenado pela empresa de tecnologia Fábrica Digital e pela PUC-Rio.

## 2.3 BIBLIOTECA DE DESENVOLVIMENTO - WXWIDGETS

Os programas criados em C++ exigem usabilidade e portabilidade devido a grande variedade de tipos de usuários e de sistemas operacionais existentes. A biblioteca de desenvolvimento wxWidgets atende essa demanda, possibilitando desenvolver aplicações multi-plataforma.

Widget, termo sem tradução que se popularizou nos últimos tempos, pode ser definido como um componente de software que viabiliza a interação com o usuário. O wxWidgets é um utilitário para criação de widgets multi-plataforma, uma biblioteca com elementos básicos para a construção de interfaces gráficas para o usuário, possibilitando conexão a bancos de dados ODBC e conectividade via sockets.

## 2.4 UML - UNIFIED MODELING LANGUAGE

A Unified Modeling Language (UML) é uma linguagem de modelagem não proprietária de terceira geração. A UML não é uma metodologia de desenvolvimento, o que significa que ela não diz para você o que fazer primeiro e em seguida ou como projetar seu sistema, mas ela lhe auxilia a visualizar seu desenho e a comunicação entre objetos.

A UML permite que desenvolvedores visualizem os produtos de seu trabalho em diagramas padronizados. Junto com uma notação gráfica, a UML também especifica significados, isto é, semântica. É uma notação independente de processos, embora o RUP (Rational Unified Process) tenha sido especificamente desenvolvido utilizando a UML.

A UML tem origem na compilação das práticas de engenharia que provaram ter sucesso na modelagem de sistemas grandes e complexos. Sucedeu aos conceitos de Booch, OMT (Rumbaugh) e OOSE (Jacobson) fundindo-os numa única linguagem de modelagem comum e largamente utilizada. A UML pretende ser a linguagem de modelagem padrão para modelar sistemas concorrentes e distribuídos.

A UML ainda não é um padrão da indústria, mas esse objetivo está a tomar forma sob os auspícios do Object Management Group (OMG). O OMG pediu informação acerca de metodologias orientadas a objetos que pudessem criar uma linguagem rigorosa de modelagem de software. Muitos líderes da indústria responderam na esperança de ajudar a criar o padrão.

Os esforços para a criação da UML tiveram início em outubro de 1994, quando Rumbaugh se juntou a Booch na Rational. Com o objetivo de unificar os métodos Booch e OMT, decorrido um ano de trabalho, foi lançado, em outubro de 1995, o esboço da versão 0.8 do Unified Process - Processo Unificado (como era conhecido). Nesta mesma época, Jacobson se associou à Rational e o escopo do projeto da UML foi expandido para incorporar o método OOSE. Nasceu então, em junho de 1996, a versão 0.9 da UML.

### **3 ANÁLISE DA SOLUÇÃO PARA O GERENCIADOR DE MÍDIAS**

#### **3.1 REQUISITOS**

##### **3.1.1 Requisitos funcionais**

RF01 – O sistema deve permitir o gerenciamento de banner (inclusão, exclusão e alteração).

RF02 – O sistema deve permitir um histórico de clique e acesso de um determinado banner.

RF03 – O sistema deve permitir o gerenciamento de clientes (inclusão, exclusão e alteração).

RF04 – O sistema deve permitir o gerenciamento de campanhas (inclusão, exclusão e alteração).

RF05 – O sistema deve permitir o gerenciamento das aplicações (inclusão, exclusão e alteração)

RF05 – O sistema deve permitir a criação dos vínculos dos banners com as campanhas.

RF06 – O sistema deve permitir a criação dos vínculos das campanhas com as aplicações.

## 3.2 CASOS DE USO

Um caso de uso descreve um objetivo que um ator externo ao sistema tem com o sistema. Um ator pode ser um elemento humano ou não que interage com o sistema. O ator se encontra fora do escopo de atuação do sistema, enquanto o conjunto de casos de uso formam o escopo do sistema. A linha que separa os atores dos casos de uso é a fronteira do sistema.

O diagrama de casos de uso representa graficamente esta interação, e define o contexto do sistema. Os atores se comunicam com os casos de uso, que é representado por uma linha unindo os dois elementos. Uma seta pode, opcionalmente, representar o fluxo principal de informação nesta interação e ajudar a leitura do caso de uso.

### 3.2.1 Descrição dos atores

Os atores representam o papel executado por uma entidade que interage com o sistema em questão. Um ator modela algo fora da fronteira do sistema que precisa trocar informações com o sistema, tais como indivíduos e outros sistemas. Uma instância pode executar o papel de vários atores diferentes e um determinado ator pode ser representado por várias instâncias.

Ator não é o mesmo que usuário. Um ator representa um papel exercido por um usuário ao interagir com um determinado caso de uso. Usuários podem desempenhar mais de um papel junto ao sistema.

Nome	Descrição
Usuário	Indivíduo não autenticado no sistema.
Administrador	Indivíduo responsável pelo gerenciamento dos clientes do sistema.
Cliente	Indivíduo responsável pelo gerenciamento dos banners e campanhas do sistema.
Visualizador	Indivíduo responsável pela visualização dos banners do sistema.

### 3.2.2 Mapa de atores

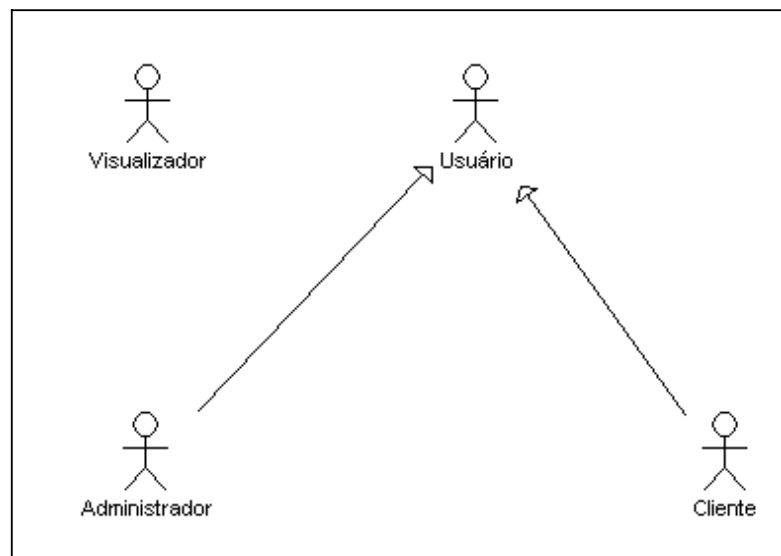


Figura 3: Mapa de atores

### 3.2.3 Diagrama de casos de uso

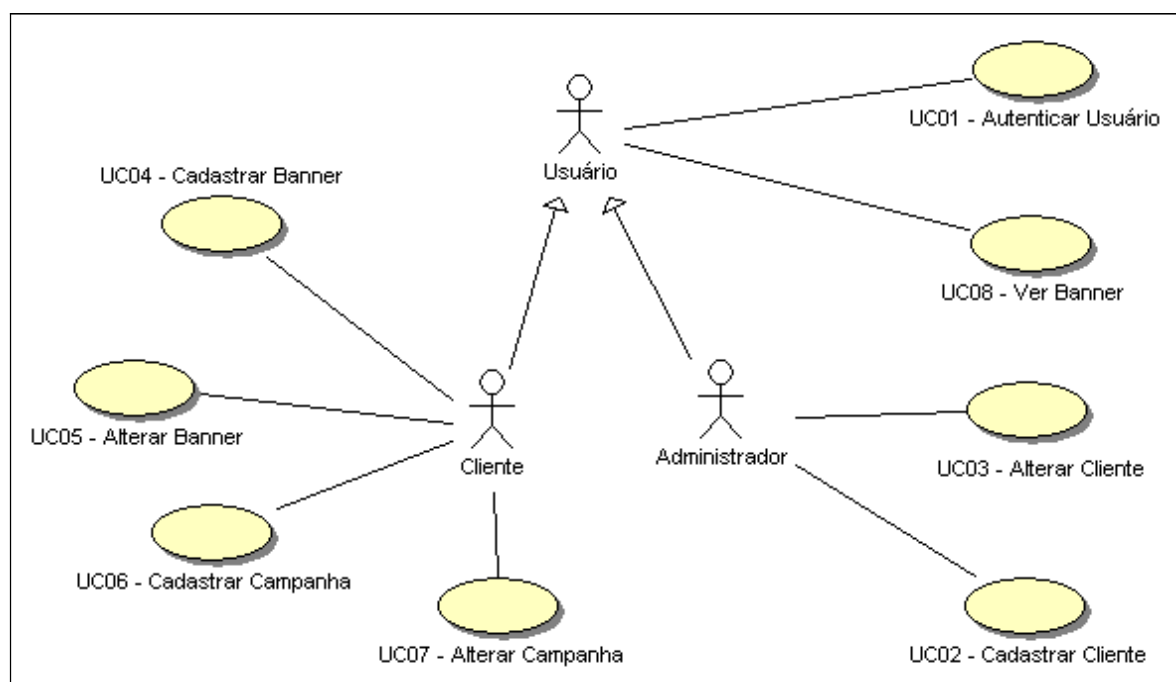


Figura 4: Diagrama de classe

### 3.2.4 Descrição dos casos de uso

#### 3.2.4.1 UC01 – Autenticar Usuário

<b>Objetivo:</b>	Permitir ao internauta se identificar e acessar as funcionalidades do sistema.
<b>Requisitos:</b>	RF15, RF16, RF18, RF19.
<b>Atores:</b>	Usuário.
<b>Pré-condições:</b>	Nenhuma.
<b>Fluxo Principal:</b>	1.O sistema solicita o email e a senha do usuário. 2.O usuário fornece o email e a senha. 3.O sistema valida a o email e a senha [A2]. 4.O sistema apresenta o menu de opções de acordo com o perfil do usuário.
<b>Fluxo Alternativo:</b>	[A2] email e/ou senha inválido 1.O sistema apresenta uma mensagem informando que email e/ou senha são inválidos e volta para o passo 1 do fluxo principal.
<b>Extensões:</b>	Não há .
<b>Pós-condições:</b>	O usuário é identificado no sistema e o menu de opções daquele usuário é apresentado.
<b>Regras de negócio:</b>	Não há. .

#### 3.2.4.2 UC02 – Cadastrar Cliente

<b>Objetivo:</b>	Permitir ao ator cadastrar um novo usuário.
<b>Requisitos:</b>	RF15, RF16, RF18, RF19.
<b>Atores:</b>	Administrador.
<b>Pré-condições:</b>	O ator deve estar autenticado no sistema.
<b>Fluxo Principal:</b>	1.O sistema solicita os dados do novo cliente: - Nome - Email - Senha - Créditos - Prioridade - Estágio - Nível 2.O ator fornece dos dados solicitados. 3.O sistema valida os dados fornecidos [A1] [RN1, RN2]. 4.O sistema registra o novo cliente.
<b>Fluxo Alternativo:</b>	[A1] Email do cliente já cadastrado: 1.O sistema apresenta uma mensagem informando que o email cadastrado já existe e volta ao passo 1 do fluxo principal.

<b>Extensões:</b>	Não há
<b>Pós-condições:</b>	O usuário é identificado no sistema e o menu de opções daquele usuário é apresentado.
<b>Regras de negócio:</b>	[RN1] Nome, email, senha, prioridade, estágio e nível são obrigatórios. [RN2] Não podem existir dois clientes com o mesmo email.

### 3.2.4.3UC03 – Alterar Cliente

<b>Objetivo:</b>	Permitir ao ator alterar os dados de um cliente.
<b>Requisitos:</b>	RF15, RF16, RF18, RF19.
<b>Atores:</b>	Administrador.
<b>Pré-condições:</b>	O ator deve estar autenticado no sistema.
<b>Fluxo Principal:</b>	1.O sistema solicita os dados do novo cliente: - Nome - Email - Senha - Créditos - Prioridade - Estágio - Nível 2.O ator altera dos dados solicitados. 3.O sistema valida os dados fornecidos [A1] [RN1, RN2]. 4.O sistema registra a alteração dos dados do cliente.
<b>Fluxo Alternativo:</b>	[A1] Email do cliente já cadastrado: 1.O sistema apresenta uma mensagem informando que o email cadastrado já existe e volta ao passo 1 do fluxo principal.
<b>Extensões:</b>	Não há
<b>Pós-condições:</b>	O usuário é identificado no sistema e o menu de opções daquele usuário é apresentado.
<b>Regras de negócio:</b>	[RN1] Nome, email, senha, prioridade, estágio e nível são obrigatórios. [RN2] Não podem existir dois clientes com o mesmo email.

### 3.2.4.4UC04 – Cadastrar Banner

<b>Objetivo:</b>	Permitir ao ator cadastrar um novo banner.
<b>Requisitos:</b>	RF15, RF16, RF18, RF19.
<b>Atores:</b>	Cliente.
<b>Pré-condições:</b>	1.O ator deve estar autenticado no sistema. 2.Deve existir pelo menos uma campanha no sistema.
<b>Fluxo Principal:</b>	1.O sistema solicita os dados do novo banner: - Nome - URL



	<ul style="list-style-type: none"> <li>- Descrição</li> <li>- Data de Publicação</li> <li>- Hora de Publicação</li> <li>- Data de Expiração</li> <li>- Hora de Expiração</li> <li>- Campanha</li> </ul> <p>2.O ator fornece dos dados solicitados. 3.O sistema valida os dados fornecidos [A1] [RN1, RN2]. 4.O sistema registra o novo banner.</p>
<b>Fluxo Alternativo:</b>	<p>[A1] Banner já cadastrado: 1.O sistema apresenta uma mensagem informando que o banner cadastrado já existe e volta ao passo 1 do fluxo principal.</p>
<b>Extensões:</b>	Não há
<b>Pós-condições:</b>	O usuário é identificado no sistema e o menu de opções daquele usuário é apresentado.
<b>Regras de negócio:</b>	<p>[RN1] Nome, data de publicação e campanha são obrigatórios. [RN2] Não podem existir dois banners com o mesmo nome.</p>

### 3.2.4.5UC05 – Alterar Banner

<b>Objetivo:</b>	Permitir ao ator alterar os dados de um banner.
<b>Requisitos:</b>	RF15, RF16, RF18, RF19.
<b>Atores:</b>	Cliente.
<b>Pré-condições:</b>	O ator deve estar autenticado no sistema.
<b>Fluxo Principal:</b>	<p>1.O sistema solicita os dados do novo banner:</p> <ul style="list-style-type: none"> <li>- Nome</li> <li>- URL</li> <li>- Descrição</li> <li>- Data de Publicação</li> <li>- Hora de Publicação</li> <li>- Data de Expiração</li> <li>- Hora de Expiração</li> <li>- Campanha</li> </ul> <p>2.O ator fornece dos dados solicitados. 3.O sistema valida os dados fornecidos [A1] [RN1, RN2, RN3]. 4.O sistema registra o novo banner.</p>
<b>Fluxo Alternativo:</b>	<p>[A1] Banner já cadastrado: 1.O sistema apresenta uma mensagem informando que o banner cadastrado já existe e volta ao passo 1 do fluxo principal.</p>
<b>Extensões:</b>	Não há
<b>Pós-condições:</b>	O usuário é identificado no sistema e o menu de opções daquele usuário é apresentado.
<b>Regras de negócio:</b>	[RN1] Nome, data de publicação e campanha são obrigatórios.

	<p><b>[RN2]</b> Não podem existir dois banners com o mesmo nome.</p> <p><b>[RN3]</b> O nome não pode ser alterado.</p>
--	--

#### 3.2.4.6UC06 – Cadastrar Campanha

<b>Objetivo:</b>	Permitir ao ator cadastrar uma nova campanha.
<b>Requisitos:</b>	RF15, RF16, RF18, RF19.
<b>Atores:</b>	Cliente.
<b>Pré-condições:</b>	O ator deve estar autenticado no sistema.
<b>Fluxo Principal:</b>	<p>1.O sistema solicita os dados da nova campanha:</p> <ul style="list-style-type: none"> <li>- Nome</li> <li>- Créditos</li> <li>- Descrição</li> <li>- Data de Publicação</li> <li>- Hora de Publicação</li> <li>- Data de Expiração</li> <li>- Hora de Expiração</li> <li>- Estágio</li> </ul> <p>2.O ator fornece dos dados solicitados.</p> <p>3.O sistema valida os dados fornecidos <b>[A1]</b> <b>[RN1, RN2]</b>.</p> <p>4.O sistema registra a nova campanha.</p>
<b>Fluxo Alternativo:</b>	<p><b>[A1]</b> Campanha já cadastrada:</p> <p>1.O sistema apresenta uma mensagem informando que a campanha cadastrada já existe e volta ao passo 1 do fluxo principal.</p>
<b>Extensões:</b>	Não há
<b>Pós-condições:</b>	O usuário é identificado no sistema e o menu de opções daquele usuário é apresentado.
<b>Regras de negócio:</b>	<p><b>[RN1]</b> Nome, créditos e data de publicação são obrigatórios.</p> <p><b>[RN2]</b> Não podem existir duas campanhas com o mesmo nome.</p>

#### 3.2.4.7UC07 – Alterar Campanha

<b>Objetivo:</b>	Permitir ao ator alterar os dados de uma campanha.
<b>Requisitos:</b>	RF15, RF16, RF18, RF19.
<b>Atores:</b>	Cliente.
<b>Pré-condições:</b>	O ator deve estar autenticado no sistema.
<b>Fluxo Principal:</b>	<p>1.O sistema solicita os dados da campanha:</p> <ul style="list-style-type: none"> <li>- Nome</li> <li>- Créditos</li> <li>- Descrição</li> </ul>

	<ul style="list-style-type: none"> <li>- Data de Publicação</li> <li>- Hora de Publicação</li> <li>- Data de Expiração</li> <li>- Hora de Expiração</li> <li>- Estágio</li> </ul> <p>2.O ator fornece dos dados solicitados.</p> <p>3.O sistema valida os dados fornecidos [A1] [RN1, RN2].</p> <p>4.O sistema registra a alteração dos dados da campanha.</p>
<b>Fluxo Alternativo:</b>	<p>[A1] Campanha já cadastrado:</p> <p>1.O sistema apresenta uma mensagem informando que a campanha cadastrada já existe e volta ao passo 1 do fluxo principal.</p>
<b>Extensões:</b>	Não há
<b>Pós-condições:</b>	O usuário é identificado no sistema e o menu de opções daquele usuário é apresentado.
<b>Regras de negócio:</b>	<p>[RN1] Nome, créditos e data de publicação são obrigatórios.</p> <p>[RN2] Não podem existir duas campanhas com o mesmo nome.</p>

### 3.2.4.8UC08 – Ver Banner

<b>Objetivo:</b>	Permitir ao ator visualizar um banner.
<b>Requisitos:</b>	RF15, RF16, RF18, RF19.
<b>Atores:</b>	Visualizador.
<b>Pré-condições:</b>	Não há.
<b>Fluxo Principal:</b>	<p>1.O ator acessa um endereço de internet do sistema.</p> <p>2.O sistema retorna os dados de um banner para o ator [A1].</p> <p>3.O sistema registra o nome do banner, a ação (VER ou CLICAR), e o protocolo de internet IP do ator.</p>
<b>Fluxo Alternativo:</b>	<p>[A1] Sem banner:</p> <p>1.O sistema apresenta uma mensagem informando que o sistema não possui banners cadastrados.</p>
<b>Extensões:</b>	Não há
<b>Pós-condições:</b>	Não há.
<b>Regras de negócio:</b>	Não há.

### 3.2.5 Diagrama de classes

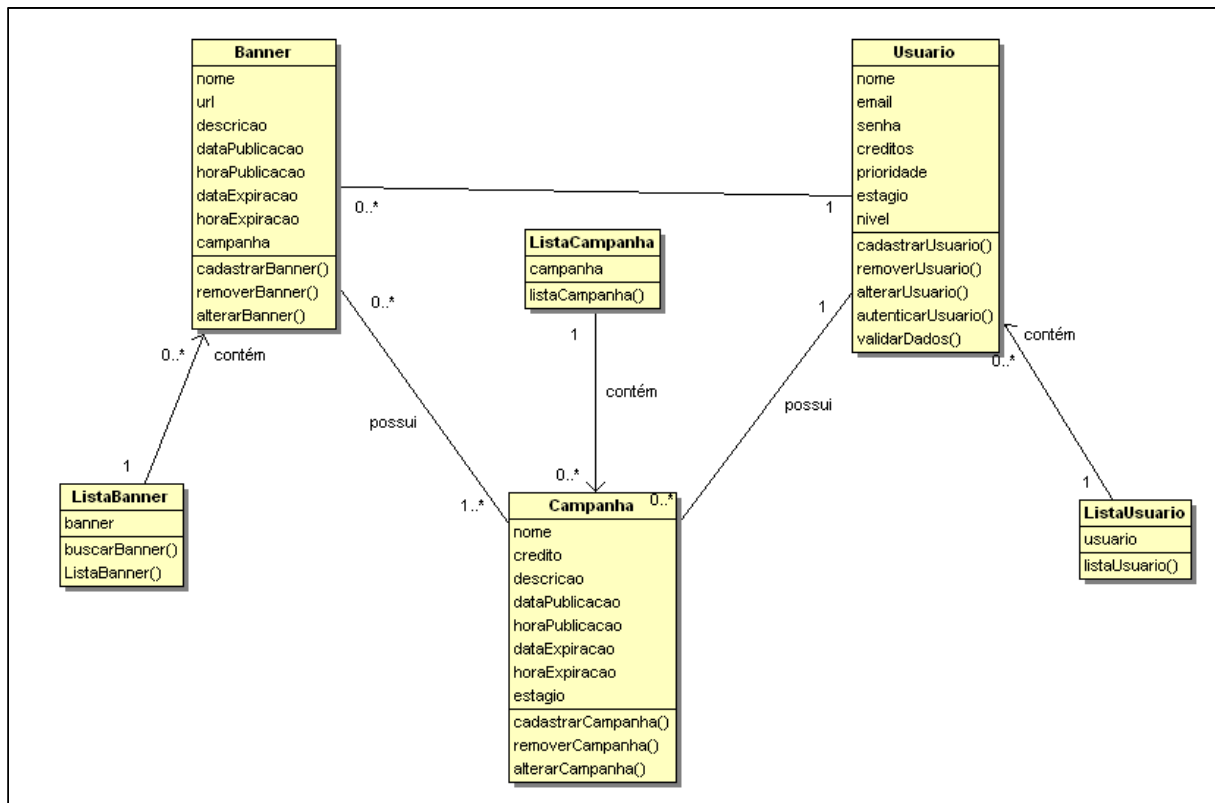
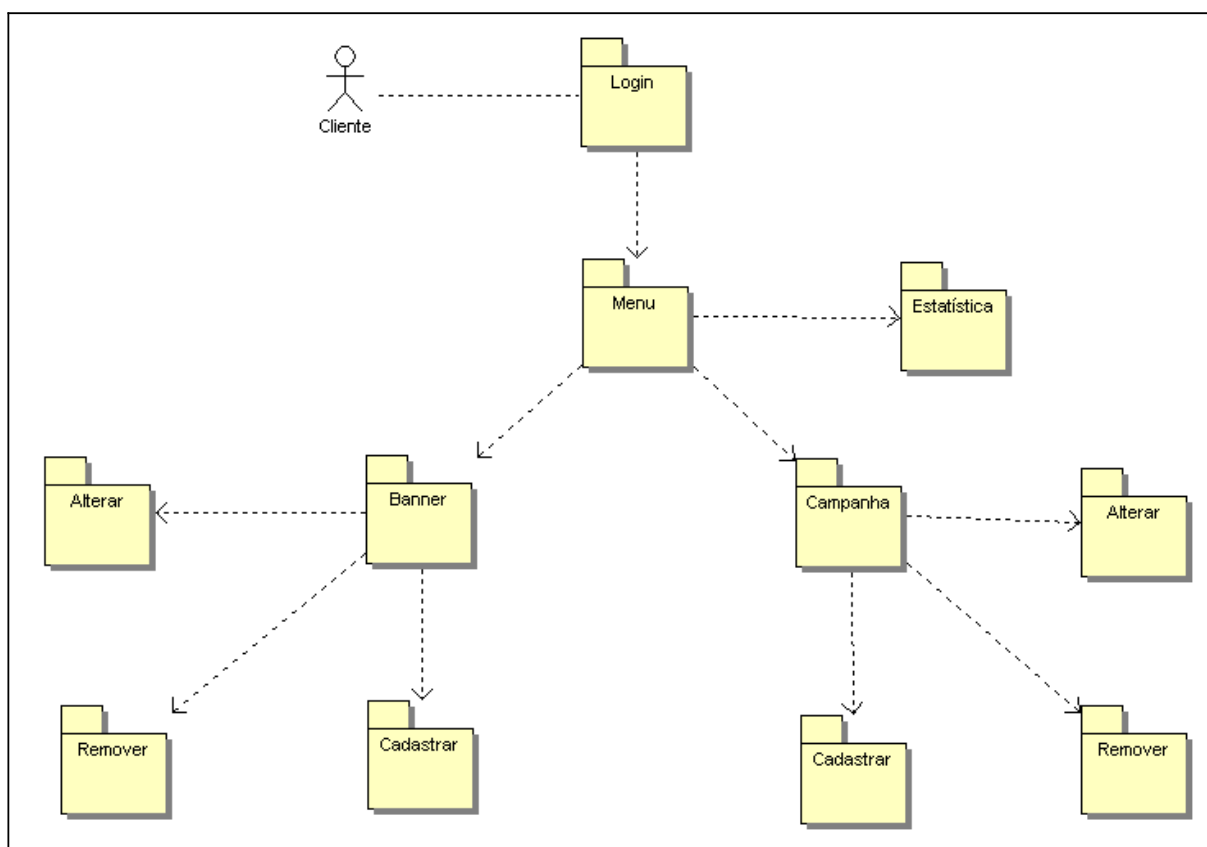


Figura 5: Diagrama de classe

Os diagramas de classe descrevem as classes que formam a estrutura do sistema e suas relações. As relações entre as classe podem ser associações, agregações ou heranças. As classes possuem além de um nome, os atributos e as operações que desempenham para o sistema. Uma relação indica um tipo de dependência entre as classes, essa dependência pode ser forte com no caso da herança ou da agregação ou mais fraca como no caso da associação, mas indicam que as classe relacionadas cooperam de alguma forma para cumprir um objetivo para o sistema.

### 3.2.6 Mapa de navegação - Cliente



**Figura 6: Mapa de navegação**

### 3.2.7 Mapa de navegação – Administrador

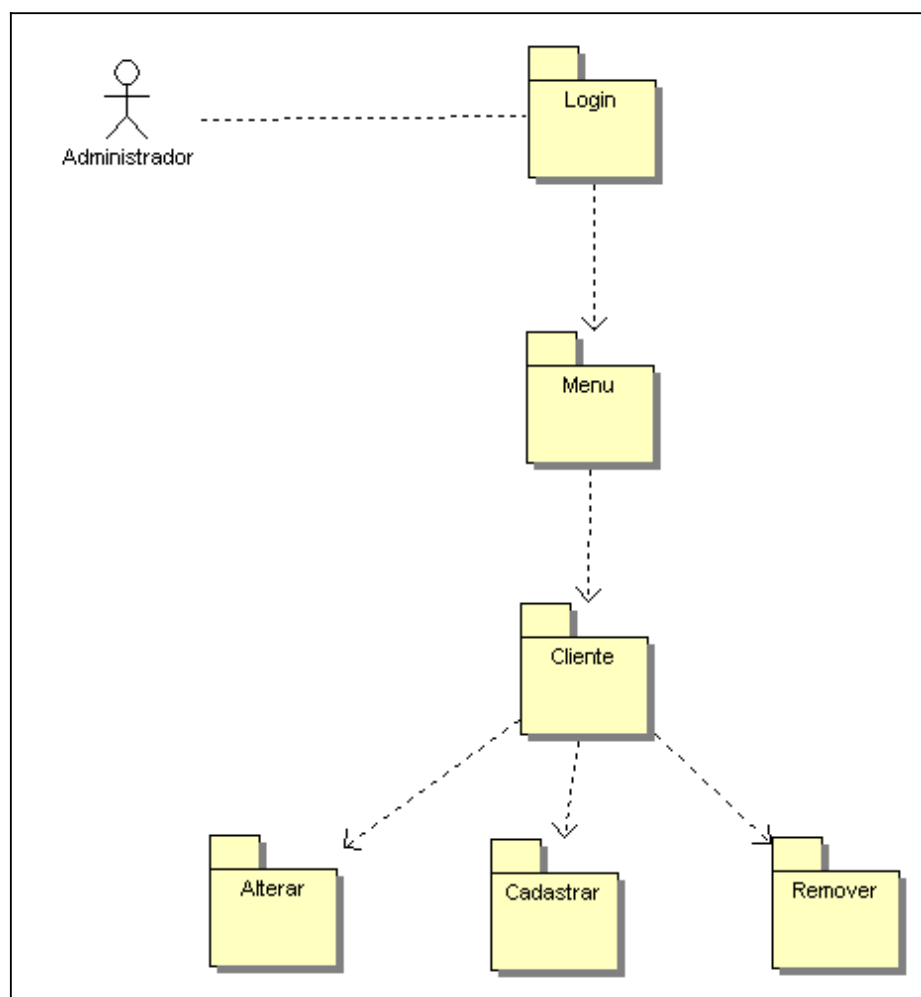


Figura 7: Mapa de navegação - Administrador

## 3.3 O SISTEMA

## 4 CONCLUSÃO

Com este projeto, mostrar que o sistema resolve o problema dos clientes que podem focar no proprio soft e utilizar o modelo de adware pronto.

## REFERÊNCIAS