

*Projeto*

*Anti Furto  
Company*

*- 2014 -*

# *Anti Furto Company*

*Linguagem para  
Desenvolvimento Web  
&  
Sistema de Gerenciamento  
de Banco de Dados*

## *II Módulo de Web*

*Luiz Augusto Oliveira de Mattos*

*nº: 1400614*

*Guilherme Kodi Abe*

*nº: 1400771*

# Sumário

<i>Introdução</i> .....	4
<i>Contextualização</i> .....	5
<i>Desenvolvimento do Banco de Dados</i> .....	6
<i>TABLES</i> .....	8
<i>VIEW</i> .....	9
<i>TRIGGER</i> .....	10
<i>INSERT</i> .....	11
<i>CREATE USERS</i> .....	12
<i>PRIVILEGIOS</i> .....	12
<i>Desenvolvimento do Sistema WEB</i> .....	13
<i>Dificuldades no desenvolvimento do banco de dados</i> .....	18
<i>Dificuldades no desenvolvimento do Sistema WEB</i> .....	19
<i>Conclusão</i> .....	20
<i>Bibliografia</i> .....	21

## *Introdução*

Anti-FurtoCompany é um site de vendas de produtos de segurança.

Anti-FurtoCompany expressa claramente sua eficiência ao transmitir aos seus leitores total segurança em relação aos seus bens (carros, motos, bicicletas, casas).

Nós Ofereceremos produtos de segurança para que as pessoas não tenham medo de roubarem os seus bens matérias.

## *Contextualização*

De acordo com o G1-noticias no ano de 2014, por dia são roubados mais ou menos 5.000 mil carros, casas, motos e bicicleta.

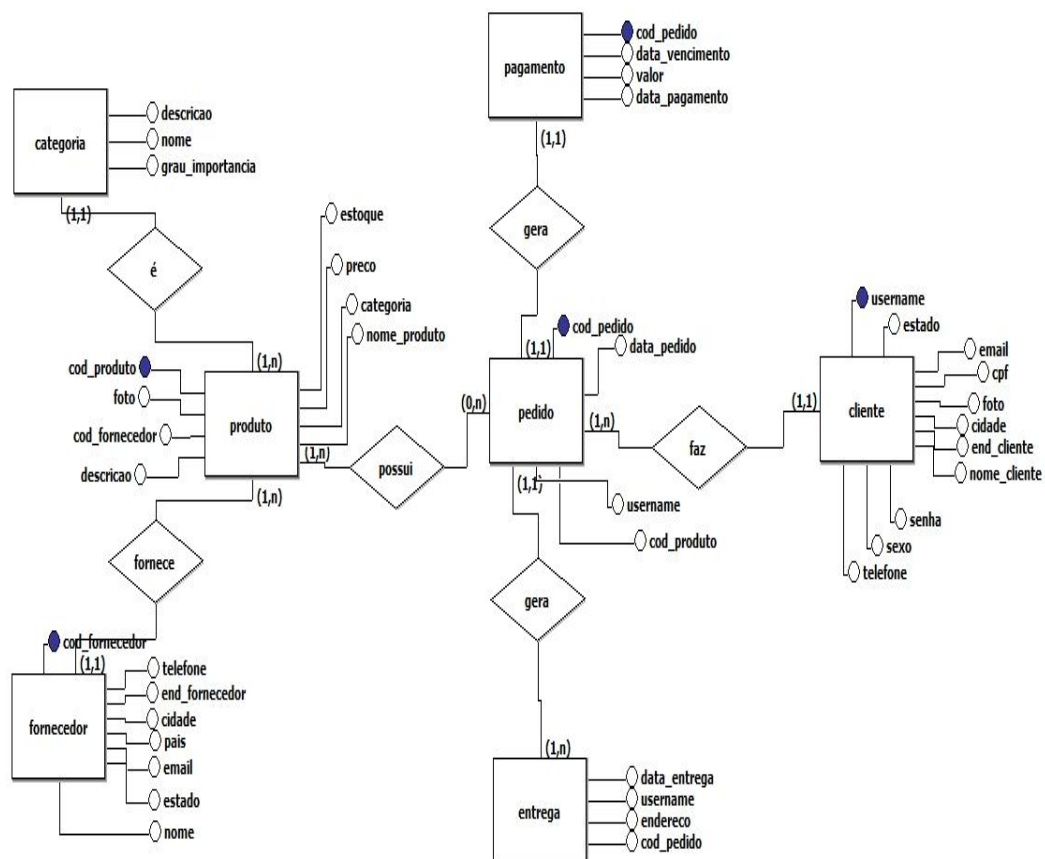
As pessoas estão deixando de sair de suas casas para se sentirem mais seguras, e com menos medo de roubarem os seus bens, mas mesmo em suas casas elas não estão tão seguras quanto elas pensam, por que se elas não tiverem segurança em suas casas, logo elas podem ser roubadas e às vezes por um pouco de dinheiro gasto na segurança de suas casas, as deixaram seguras.

Dois Alunos do IFPS (Instituto Federal de São Paulo) Luiz Mattos e Guilherme Kodi, elaboraram um site de segurança chamado Anti furto Company, com o propósito de dar mais segurança as pessoas e tentar diminuir o índice de roubo no Brasil.

## *Desenvolvimento do Banco de Dados*

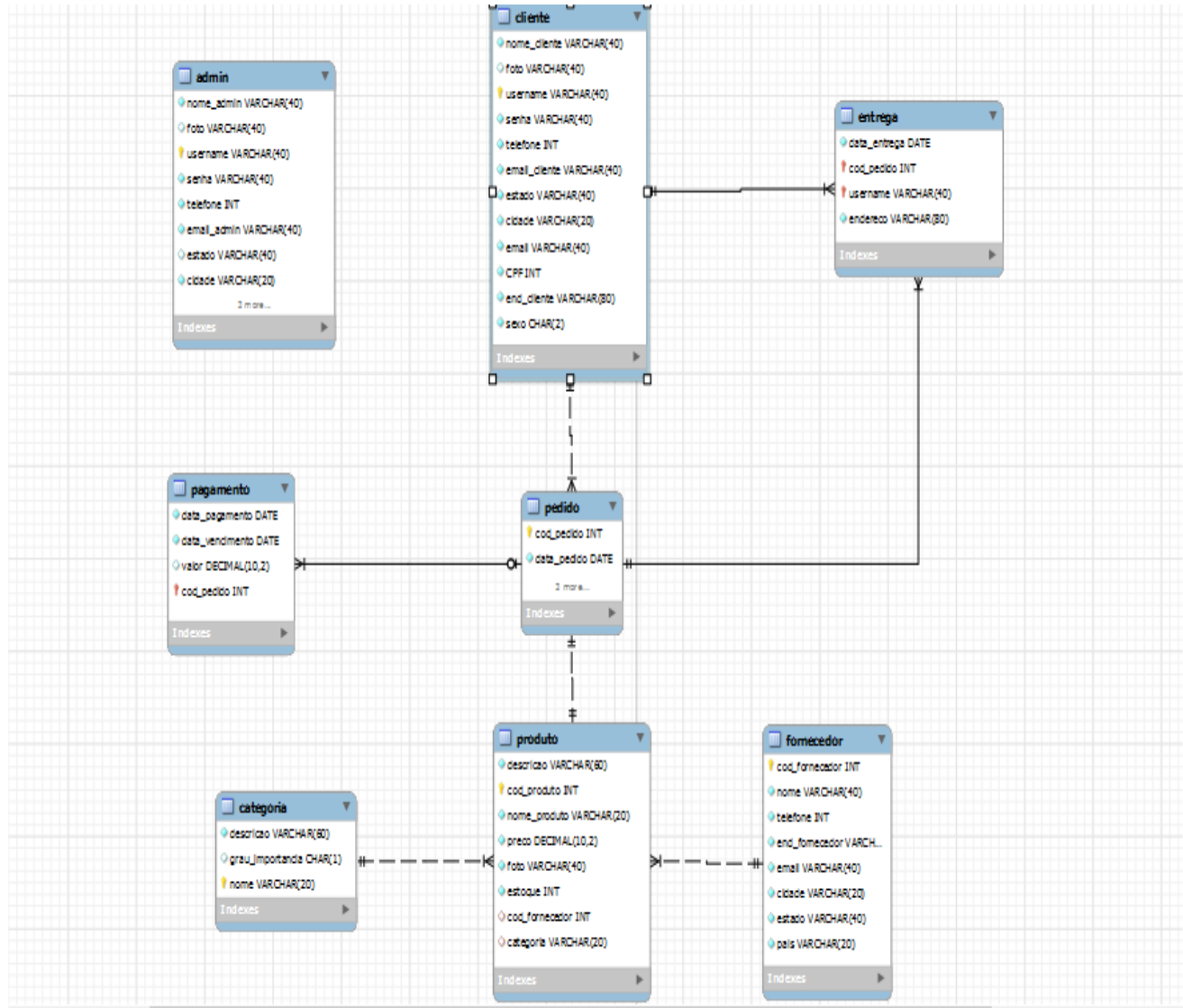
Já com o MER pronto, uma reunião foi marcada, na casa do Luiz Augusto, para o desenvolvimento do MRel. Com o desenvolvimento do MRel, foi criada linhas de código sql no Mysql, para a criação do banco de dados.

Abaixo está o MER



Foram criadas 5 entidades, são elas: pedido, produto, cliente, fornecedor, entrega, categoria e pagamento. Onde um produto irá possuir apenas uma categoria e seria fornecido por apenas um fornecedor, logo um fornecedor poderia fornecer vários produtos de diversas categorias. Um pedido seria gerado por um cliente, logo um cliente poderia fazer vários pedidos diferentes. Resultando em um pagamento e uma entrega do pedido.

Abaixo está o MRel



MRel desenvolvido em base no MER.

## ***TABLES***

Clientes possuem como chave primaria o username, assim tendo que ser único, para ser também utilizado como item de busca.

Fornecedores possuem como chave primaria cod\_fornecedor, assim tendo que ser único, para poder cadastrar os produtos.

Produtos possuem como chave primaria cod\_produto, assim tendo que ser único, para poder mandar os produtos para pedido.

Categorias possuem como chave primaria nome, para poder colocar os itens em sua classe.

Pagamento possui como chave estrangeira cod\_pedido, para poder colocar pegar o pedido que foi feito e seu preço.

Entrega possui como chaves estrangeiras cod\_pedido e username, para poder pegar os dados do cliente, para fazer a entrega.

Com a finalização do MRel, foi criado tabelas no Myqsl seguindo o MER e o MRel.

Foi criada a tabela *admin*, onde seria armazenado os administradores do site, mantendo controle de acesso sobre determinado local do site, isso implementado em PHP.

Também foi criada a tabelahistorico\_update, onde seria armazena a ação, o produto alterado e qual usuário do banco de dados o altero.



## ***VIEW***

Inicialmente foi criada apenas uma View.

```
CREATE VIEW tabela_cliente as select foto, username, nome_cliente, sexo, email_cliente from cliente;
```

Esta View tem como função limitar o acesso de dados da tabela cliente para os usuários do site, assim não revelando determinados dados pessoais, como a senha, CPF, endereço, estado e o número de telefone.

Logo depois foi decidido o desenvolvimento de mais uma View, para o controle de pedidos.

```
CREATE VIEW tabela_pedido as select pe.cod_pedido, pe.data_pedido,
                                c.username, c.nome_cliente,
                                pr.cod_produto, pr.nome_produto,
                                f.nome
                                from pedido as pe
INNER JOIN (cliente as c inner join (produto as pr inner join fornecedor as f))
ON pe.username = (c.username AND pe.cod_produto = pr.cod_produto);
```

Essa View tem como função facilitar a visualização dos pedidos feitos pelos clientes do site, pelo administrador. Com essa View pode ser visualizado com apenas um chamado, dados de 4 tabelas diferentes de uma só vez. São elas: cliente, produto, fornecedor, pedido.

## ***TRIGGER***

Foi criada duas Triggers com o objetivo de inserir dados na tabela *historico\_update*, para controlar as ações feitas na tabela produtos, funcionaria como um histórico de alterações, onde armazenaria qual produto, valor antigo, valor novo, alterado ou excluído e por qual usuário de banco.

```
DELIMITER $
CREATE TRIGGER update_historico_produtos AFTER UPDATE on produto FOR EACH ROW
BEGIN
    INSERT INTO historico_update (cod_produto, acao, user, valor_old, valor_new)
        VALUES (OLD.cod_produto, "update", user(), OLD.preco, NEW.preco);
END $
DELIMITER ;
```

Nessa Trigger é armazenado na tabela *historico\_update*, o produto alterado, preço antigo, preço novo, a ação “alterar” e o usuário de banco que alterou, cada vez que um update é feito na tabela produtos.

```
DELIMITER $
CREATE TRIGGER update_historico_produtos_delete AFTER DELETE on produto FOR EACH ROW
BEGIN
    INSERT INTO historico_update (cod_produto, acao, user)
        VALUES (OLD.cod_produto, "delete", user());
END $
DELIMITER ;
```

Nessa segunda Trigger é armazenado na tabela *historico\_update*, o produto excluído, a ação “delete” e o usuário de banco que deletou, cada vez que um delete é feito na tabela produtos.

## ***INSERT***

Alguns dados já foram inseridos nas tabelas como padrão, como as categorias para os produtos.

```
INSERT INTO categoria (nome, descricao, grau_importancia)
VALUES ("Bicicleta", "Produtos anti-furto para bicicletas" , "1");
INSERT INTO categoria (nome, descricao, grau_importancia)
VALUES ("Casa", "Produtos anti-furto para casas", "2");
INSERT INTO categoria (nome, descricao, grau_importancia)
VALUES ("Carro", "Produtos anti-furto para carros", "2");
INSERT INTO categoria (nome, descricao, grau_importancia)
VALUES ("Moto", "Produtos anti-furto para motos", "2");
```

Categorias Bicicleta, Casa, Carro e Moto como padrão para os produtos vendidos na loja.

Também foi criado o primeiro administrador na tabela admin, onde pode ser alterado futuramente, conforme a necessidade do desenvolvedor.

```
INSERT INTO admin (nome_admin, username, senha, email_admin)
VALUES ("Administrador", "admin", "admin", "adm_antifurto@gmail.com");
```

## ***CREATE USERS***

Foi criado 3 tipos de users para o banco de dados.

```
CREATE USER webmaster@localhost  
IDENTIFIED BY "webmaster";  
  
CREATE USER cliente@localhost  
IDENTIFIED BY "cliente";  
  
CREATE USER visitante@localhost  
IDENTIFIED BY "visitante";
```

## ***PRIVILEGIOS***

Para cada user, existiria determinado nível de permissão, onde o user webmaster, teria acesso total de todo o bando de dados.

O user cliente teria permissão para visualizar as tabelas: cliente, produto, histórico\_update, entrega, fornecedor, pagamento, categoria e pedido. Permissão para atualizar as tabelas: cliente, produto, histórico\_update, entrega, pagamento e pedido. Permissão para inserir dados nas tabelas: cliente, produto, histórico\_update, entrega, pedido e pagamento.

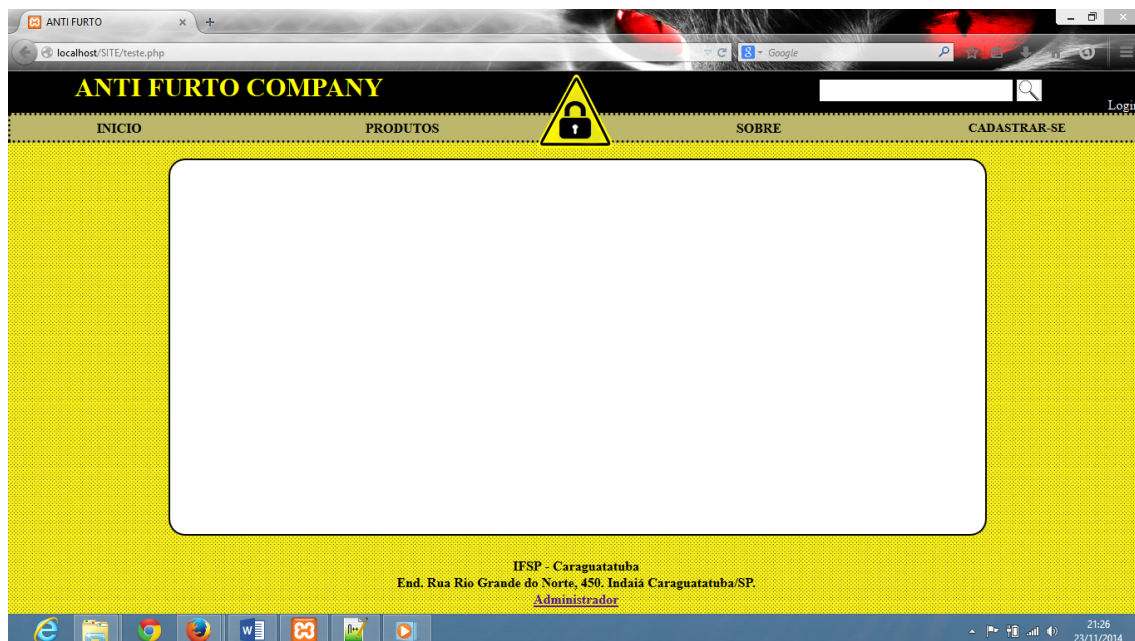
Já o user visitante teria acesso apenas as tabelas: produto, fornecedor, categoria, admin, cliente e apenas inserir dados na tabela cliente, para seu cadastro.

## *Desenvolvimento do Sistema WEB*

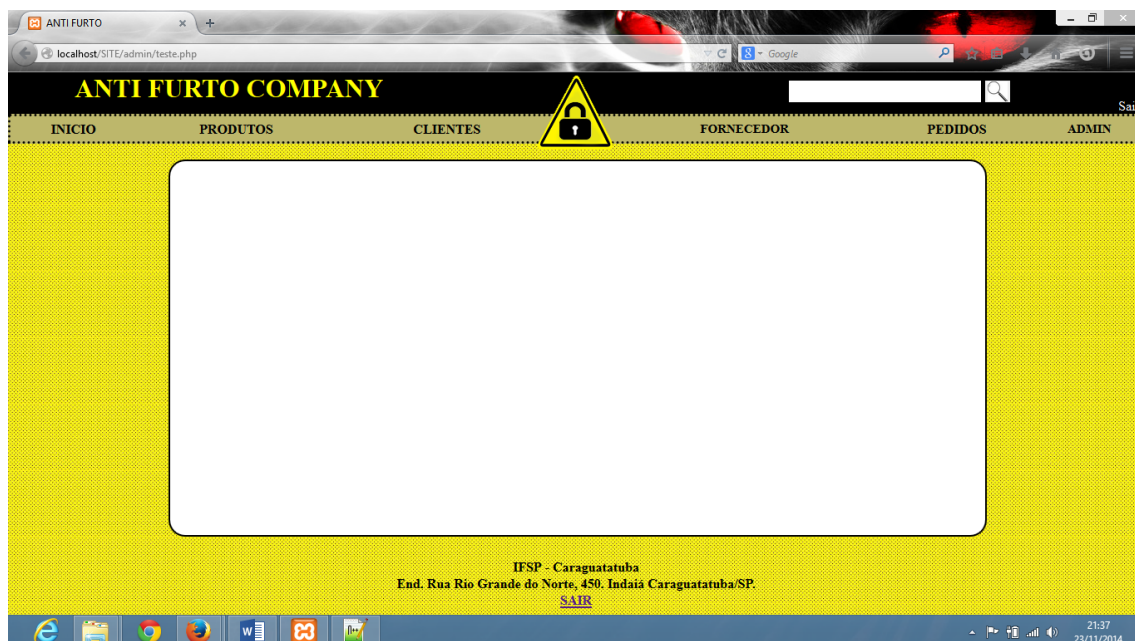
Após ser desenhado em um papel o design do site como rascunho, começamos a criação das divs principais para a estrutura do site.

Utilizado como padrão um único arquivo CSS, onde conteria todas as formatações para todas as páginas do site. Assim sendo criada 5 divs principais, são elas: wrapper(div que segura as demais divs), header(cabeçalho), nav(menu), content(conteúdo) e por último a div footer(rodapé).

Assim criando o template do site, uma estrutura padrão que seria utilizado em todas as páginas do site, usando o comando “include”.



Acima está o template para as páginas apresentadas para os usuários do site.



Acima está o template para as páginas apresentadas para os administradores do site.

Os links disponíveis para ambos são diferentes, pois cada um terá uma limitação de acesso aos dados.

Para acessar tais partes do site, será necessário cadastrar e estar logado. Obtendo uma conexão, com um outro arquivo php, com o bando de dados, é feito a consulta no bando e inserindo o novo cliente. O cliente para se cadastrar, deverá preencher os campos. Esses dados são enviados e armazenados na tabela cliente.

ANTI FURTO COMPANY

INICIO PRODUTOS SOBRE CADASTRAR-SE

CADASTRO

Nome:

Sexo: ☒ Masculino ☐ Feminino

E-mail:

Username:

Senha:

CPF:

Telefone:

Endereço:

Estado:

Cidade:

Foto:  Nenhum arquivo selecionado.



O usuário deverá informar seu username e sua senha, logo depois é verificado no bando de dados se tais dados existem e se corresponde o username com a senha. Caso seja verdadeiro, o usuário será logado no site. É armazenado em sessions os dados de login, o username e a senha, para que seja verificado em cada uma das páginas que o usuário tentar acessar.



Para manter esse controle, fui implementado nas páginas um include de uma página PHP, onde será feito a validação de login, verificando se o usuário está logado e se esse usuário existe na tabela de cliente (criada no banco de dados). Comparando os dados armazenados nas sessions.

```
session_start();
$_SESSION['nome_usuario'] = $username;
$_SESSION['senha_usuario'] = $senha;
$_SESSION['logado'] = 'okey';
header ("location: index.php");
```

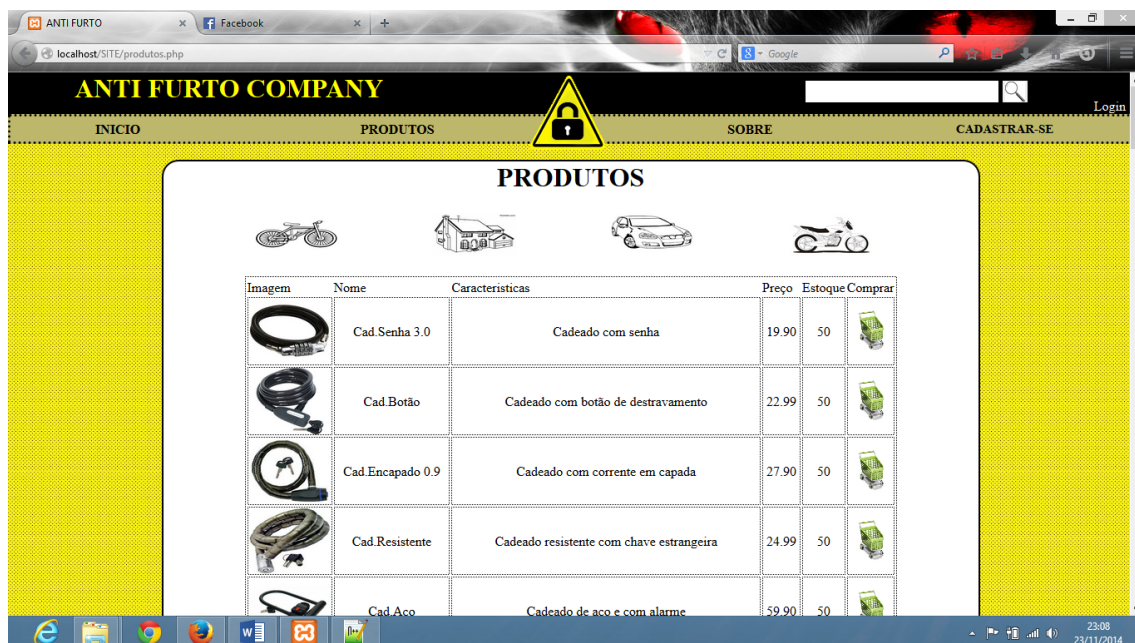
As senhas são armazenadas criptografadas no banco de dados, para que apenas o usuário tenha acesso a ela.

As imagens utilizadas pelos clientes correm o risco de obterem o mesmo nome, assim os nomes delas são armazenadas no bando de dados, criptografadas e somado a hora atual, assim uma imagem nunca terá o nome igual. As imagens são armazenadas em uma pasta chamada “fotos\_clientes”.

O mesmo vale para os produtos inseridos no site. Através da área do administrador, pode ser inserido produtos, para que seja inserido o administrador deve preencher os dados conforme as características do produto, também será obrigado a inserir o código do fornecedor do produto.

O administrador também pode inserir fornecedores no site.

Os produtos são organizados em forma de tabela. O usuário também pode selecionar a categoria que deseja no superior da tabela. Ele apenas pode comprar um produto se estiver logado.



No canto direito superior da página, encontra-se uma caixa de busca de produtos e um botão de login. O botão muda para “logout” quando o usuário estiver logado.

Ao ser finalizada a confirmação da compra, é atualizado o estoque do produto, diminuindo um em seu estoque atual. O usuário poderá apenas comprar um produto por vez, assim evitando a venda por atacado.

Caso o produto chegue ao estoque ZERO, ele não aparecerá mais para os clientes, assim evitando a compra de um produto sem estoque. Apenas os administradores poderão visualiza-lo, para que possa ser atualizado.



O login para a área do administrador se encontra no inferior de qualquer página do site, quando o administrador estiver logado, o link “Administrador” mudará para “sair”.

ANTI FURTO COMPANY

INICIO PRODUTOS SOBRE CADASTRAR-SE

### LOGIN ADMINISTRADOR

Username:

Senha:

Apagar Logar

IFSP - Caraguatatuba  
End. Rua Rio Grande do Norte, 450, Indaia Caraguatatuba/SP.  
[Administrador](#)

Um Administrador poderá visualizar os clientes e os outros administradores do site, também poderá alterar ou até mesmo em último caso excluí.

## *Dificuldades no desenvolvimento do banco de dados*

Houve dificuldades na construção das Triggers.

Primeiramente foi criado uma Trigger para que cada vez que uma compra fosse efetuada seria diminuído um do estoque. A trigger não foi criada corretamente, as condições forem pegadas erradas, assim cada vez que uma compra era efetuada era diminuído um de todos os produtos e não somente do que foi comprado.

Como não foi conseguido tratar esse problema, foi feita esse controle através do php, assim foi apagada a trigger criada.

Duas triggers foram criadas no lugar da excluída, mas agora com o objetivo de construir um histórico dos produtos alterados ou excluídos.

Houve também dificuldades na criação das tabelas com chave estrangeira. O uso não correto para chamar uma foreign key comprometeu a criação da tabela.

Logo a tabela foi criada corretamente com uma rápida pesquisa nos exemplos utilizados na sala de aula pelo professor Henrique.

## *Dificuldades no desenvolvimento do Sistema WEB*

Houve problemas com as session, onde elas não armazenavam os dados de login para que seja verificado em cada página que o usuário tentasse visualizar.

Inicialmente o projeto foi entregue com tais problemas de session, mas com a ajuda do professor Alan, o problema foi resolvido.

A solução estava na implementação incorreta das session.

Houve problemas com a div header e nav, que seriam fixas no superior da página. Elas estavam em cima da div contente, assim tampando uma parte do conteúdo. Como solução foi criado uma div chamada suporte, tinha função apenas de estar entre a div fixada e a div contente para que pudesse ser separada uma da outra.

## *Conclusão*

Com este projeto podemos concluir que para nós aprender, nós temos que praticar, por que sem a pratica não conseguimos fazer algo correto. Também podemos perceber que é muito difícil a vida de um programador, por que ele trabalha contra o tempo, e sempre que ele acha que acabou ainda tem um erro (bug). Esse trabalho nos ajudou de muitas formas principalmente quando nós achamos que era fácil, mas na verdade era muito difícil, nós conseguimos ultrapassar nossas expectativas e também podemos perceber que não existe um limite para a nossa aprendizagem.

## *Bibliografia*

- <http://g1.globo.com>
- Manual de Referência do MySQL 4.1
- [http://www.w3schools.com/sql/sql\\_view.asp](http://www.w3schools.com/sql/sql_view.asp)
- [http://wiki.locaweb.com.br/pt-br/Exemplo\\_de\\_como\\_criar\\_uma\\_trigger\\_no\\_Mysql](http://wiki.locaweb.com.br/pt-br/Exemplo_de_como_criar_uma_trigger_no_Mysql)
- [http://www.w3schools.com/php/php\\_sessions.asp](http://www.w3schools.com/php/php_sessions.asp)
- <http://www.w3schools.com/css/default.asp>