

# Delta Direct

---

## Documentação Técnica

**Rodrigues, Diego**

## Informações Técnicas

### Hospedagem

O sistema é uma aplicação web hospedada na Localweb sob o domínio [deltafaucetrep.com](http://deltafaucetrep.com). Para ter acesso ao painel de desenvolvedor basta entrar no site da Localweb e clicar no botão **Central do Cliente**, e inserir as seguintes informações de acesso:

**Usuário:** deltafaucet

**Senha:** Delta123

Uma vez obtido o acesso, para acessar o painel do desenvolvedor referente ao domínio da aplicação, basta clicar no ícone **Hospedagem de Sites** na seção **Meus Produtos**. Você será redirecionado para o **painel de hospedagem**, onde você poderá administrar os recursos da aplicação, tais como servidor ftp, banco de dados MySQL e WebMail.

A Localweb restringe certas opções root no servidor linux e no banco de dados MySQL, ou seja, você não pode excluir banco de dados, para criar um novo banco de dados você deve clicar no botão **Gerencie os Bancos**. Nesta área você terá acesso a todos banco de dados registrados no seu servidor, lembrando que estes são gerenciados pela Localweb, você não tem todas funcionalidades de um banco de dados em um servidor local próprio.

Com relação ao servidor de email, a localweb oferece em seu plano 25 contas com espaço limitado a 20GB. Para ter acesso aos emails utilizados pelo sistema basta acessar o painel de hospedagem e clicar em **Adiminstrar Emails** do domínio utilizado pela aplicação.

A aplicação utiliza o email [sistema@deltafaucetrep.com](mailto:sistema@deltafaucetrep.com) para enviar orçamentos e pedidos por email, por isso não exclua-o em hipótese alguma. Caso haja necessidade de cadastrar um novo email para um parceiro da Delta que gostaria de utilizar o domínio, por exemplo, a Myrella não conseguiria um email no domínio da Delta Internacional por ser tercerizada e não um empregado efetivo. Por isso criamos o email [myrella.honma@deltafaucetrep.com](mailto:myrella.honma@deltafaucetrep.com) para seu uso pessoal como representante da Delta. Neste caso, os email a ele enviados serão armazenados no servidor da localweb e para que ela o utilize externamente as seguintes instruções devem ser seguidas:

Acesse o painel de gerenciamento das caixas postais e clique no botão **Criar Caixa Postal** e preencher o formulario com o nome de usuário e senha. Uma vez cadastrado, basta ir no Outlook ou outra ferramenta que permita conectar-se via SMTP a servidores de emails externos e adicionar as seguintes credenciais:

**Usuário:** [email@deltafaucetrep.com](mailto:email@deltafaucetrep.com)

**Senha:** \*\*\*\*\*

#### HOSTS

**IMAP:** email-ssl.com.br (porta 993)

**POP:** email-ssl.com.br (porta 995)

**SMTP:** email-ssl.com.br (porta 465)

## Servidor

O servidor é hospedado numa máquina linux com acesso restrito a pasta **/home/deltafaucetrep1**, que é a raiz da aplicação. Para ter acesso via FTP aos arquivos do servidor basta utilizar um cliente FTP qualquer e inserir as seguintes credenciais:

**Host:** ftp.deltafaucetrep.com

**Porta:** 21

**Senha:** Delta123

A aplicação localiza-se na pasta **/public\_html** e é organizada da seguinte maneira:

**/public\_html** : pasta raiz da aplicação

**/app** : pasta com o código da aplicação

**/Console** : classes responsáveis por acessar e manipular o console

**/Events** : classes responsáveis por manipular eventos

**/Exceptions** : classes responsáveis por lidar com erros ocorridos no sistema

**/Http** : classes responsáveis por lidar com requisições Http

**/Controllers** : controladoras de requisições da aplicação

**/Auth**: controladoras responsáveis por validar sessões de usuários

**/Middleware** : intermediários de requisições da aplicação

**/Request** : classes responsáveis por conter requisições

**Kernel.php** : classe que cadastra e libera acesso aos intermediário

**routes.php** : mapeamento entre requisições e controladoras

**/Jobs** : classes abstrata para conter jobs internos do servidor, não utilizada

**/Listeners** : classes que ficam na escuta por eventos disparados pelas controladoras

**/Providers** : classes provedoras de serviços para controladoras e views

**User.php** : classe modelo para conter informações do atual usuário usando a aplicação

... : demais modelos, para cada tabela no banco existe um modelo aqui

**/bootstrap** : arquivos de inicialização da framework Laravel

**/config** : arquivos de configuração da aplicação

**app.php** : arquivo principal de configuração, mapeando a regionalidade e as classes do sistema

**auth.php** : arquivo de configuração do driver de autenticação

**broadcasting.php** : arquivo de configuração para uso de websockets, não utilizados

**cache.php** : arquivo de configuração do cache do servidor

**compile.php** : arquivo com arquivos para serem pré-compilados

**database.php** : arquivo de configuração entre a aplicação e o banco de dados

**filesystems.php** : arquivo de configuração do servidor FTP

**mail.php** : arquivo de configuração do email linkado ao servidor da localweb

**queue.php** : arquivo de configuração da pilha de tarefas do servidor

**services.php** : arquivo de mapeamento dos serviços internos do servidor

**session.php** : arquivo de configuração das sessions do servidor

**view.php** : arquivo de configuração do sistema de views do servidor

**/database** : scripts e classes relacionadas a manutenção do banco de dados

**/factories** : fabricas de classe para semeamento do banco de dados

**/migrations** : arquivos para efetuar migrações no banco de dados (Laravel migrations api)

**/scripts** : scripts usados durante a vida do banco, em ordem de execução

**/0.0.1**: scripts de criação do banco de dados

**create\_tables.sql** : script de criação de todas tabelas do banco de dados

**headache.pdf** : diagrama do banco de dados

**run\_this\_script.sql** : script de bootstrap do banco de dados

... : scripts em ordem de execução

**/public** : pasta de acesso da aplicação, todos request ao dominio deltafaucetrep.com serão redirecionados a esta pasta

**index.php** : inicia o bootstrap da aplicação

**robot.txt** : arquivo para configurações de SEO

**.htaccess** : sobscrive os request para se adequarem ao funcionamento da aplicação

**/resources** : recursos usados pela aplicação

**/lang** : arquivos para internacionalização da aplicação

**/views** : views e partials da interface da aplicação

**/auth:** telas de autenticação e login

**/layouts:** páginas masters da aplicação

**/... :** outras telas parciais da aplicação, todas estendem das masters

**index.blade.php :** tela inicial da aplicação

**/static\_files :** arquivos estáticos utilizados pelas telas da aplicação (**sub-dominio:** static.deltafaucetrep.com)

**/assets:** arquivos JS, CSS e imagens usadas pela interface da aplicação

**/avatars:** imagens de perfil dos usuários

**/commercial:** arquivos de uso comercial fornecidos aos usuários

**/police :** políticas comerciais divulgadas pela gerência comercial

**/price\_list :** listas de preços fornecidas pela Delta

**/technologies:** imagens das tecnologias da Delta, usadas no guia de especificação online

**/tutorial:** print-screens utilizados nos tutoriais na seção de ajuda da aplicação

**/storage :** pasta para arquivos que foram enviados ao servidor pelos usuários

**/app :** arquivos de manutenção da aplicação, não utilizados

**/files :** arquivos que foram gerados pelos serviço CloudStorage

**/inventory:** estoques que foram enviados pelos distribuidores

**/price\_police:** políticas de preços enviadas por distribuidores

**/framework :** arquivos gerados pelo funcionamento do servidor (Não mexer)

**/cache:** informações sobre o cache do servidor

**/sessions:** informações sobre sessões abertas no servidor (**criptografadas**)

**/views:** informações para otimização da renderização de views

**/logs :** log da aplicação (Não abrir)

**/orders :** arquivos de faturamento dos subpedidos, recomendo fazer backup semanal

**/users :** arquivos de usuários e análises de crédito, recomendo fazer backup semanal

**/tests :** arquivos com testes unitários automáticos, não foram utilizados ☹

**/vendor :** bibliotecas e apis de terceiros usados pela aplicação

**.env :** arquivo de configuração do ambiente, cuidado!

**.htaccess :** arquivo de redirecionamento do servidor apache, cuidado!

**server.php** : arquivo de auxilio ao redirecionamento do servidor apache, cuidado!

... : repositório git, licença e outros arquivos de IDE

Os arquivos de código do sistema não devem ser alterados sem o acompanhamento do criador, pois estes são muito sensíveis e podem acarretar o mal-funcionamento da aplicação. Caso uma alteração seja feita e um erro exploda, consulte o repositório no github, [github.com/diego95](https://github.com/diego95) para verificar o versionamento da aplicação.

## Banco de Dados

O banco de dados utilizado pela aplicação é o MySQL devido o amplo suporte oferecido pelo Host, incluindo backups e acesso ao **PhpMyAdmin** que possibilita um fácil gerenciamento do banco. Lembrando que a Localweb restringe o acesso root ao banco de dados, apenas um usuário definido durante a criação do mesmo.

Para acessar o banco de dados principal da aplicação vá ao painel de gerenciamento de banco de dados da aplicação e clique no botão **adminstrar** no banco de dados nomeado **deltafaucetrep11**, aí basta inserir a senha **Delta123** e você será redirecionado ao PhpMyAdmin fornecido pela localweb.

Para acessar o banco de dados remotamente basta inserir as seguintes credenciais:

**Host:** mysql10.deltafaucetrep1.hospedagemdesites.ws

**Usuário:** deltafaucetrep11

**Senha:** Delta123

**Banco de dados:** deltafaucetrep11

A aplicação faz uso de 113 tabelas sendo:

**user:** todos usuários do sistema estarão registrados nesta tabela

**user\_type:** são os tipos de usuários do sistema

**user\_filesystem:** relação entre usuários e arquivos enviados ao sistema

**representant:** tabela com detalhes dos usuários representantes do sistema

**representant\_portfolio:** tabela com as carteiras de clientes dos usuários representantes

**representant\_profile:** tabela com perfis de representantes

**manager:** tabela com detalhes dos usuários gerentes de vendas do sistema

**manager\_portfolio:** tabela com as carteiras de clientes dos usuários gerentes de vendas

**distribuidor:** tabela com detalhes dos usuários distribuidores do sistema

**distribuidor\_portfolio:** tabela com carteira de clientes dos usuários distribuidores

**distribuidor\_inventory\_product:** tabela com situação de produtos nos estoques dos distribuidores

**customer:** todos clientes cadastrados estão nesta tabela

**customer\_bank\_account:** contas bancárias dos clientes

**customer\_bussiness\_contact:** contatos comerciais dos clientes

**customer\_channel:** canais de distribuição possíveis dos clientes (From Sellout)

**customer\_classification:** classificações possíveis dos clientes (From Sellout)

**customer\_credit\_analysis:** análise de crédito dos clientes

**customer\_credit\_analysis\_note:** anotações da análise de crédito

**customer\_credit\_analysis\_range:** possíveis limites de créditos a serem respondidos

**customer\_credit\_analysis\_status:** possíveis estados da solicitação de análise de crédito

**customer\_credit\_analysis\_request:** pedido de análise de crédito enviado aos dists.

**customer\_credit\_analysis\_request\_note:** anotação do pedido de análise...

**customer\_credit\_analysis\_request\_status:** estados possíveis do pedido de...

**customer\_distribuidor:** distribuidor definido para atender o cliente (From Sellout)

**customer\_main\_provider:** tabela com principais fornecedores do cliente

**customer\_manager:** tabela com gerente de vendas que podem cuidar dos clientes (From Sellout)

**customer\_note:** tabela com anotações feitas no cliente

**customer\_partner:** tabela com sócios do cliente

**customer\_region:** tabela com regiões administrativa definidas pela Delta (From Sellout)

**customer\_salesman:** tabela com possíveis vendedores do cliente (From Sellout)

**customer\_sanitary\_metals\_supplier:** relação entre os fornecedores de metais sanitários e clientes

**customer\_sanitary\_ware\_supplier:** relação entre fornecedores de louças sanitárias e clientes

**customer\_segment:** possíveis segmentos dos clientes (From Sellout)

**customer\_status:** possíveis status do cliente (From Sellout)

**customer\_type:** possíveis tipos de clientes, usado no formularios de cadastro

**product:** todos os 16mil produtos da Delta estarão nesta tabela

**product\_attribute:** possíveis atributos do produto (From Int. Price List)

**product\_brand:** possíveis marcas do produto (From Int. Price List)

**product\_finish:** possíveis acabamentos dos produtos (From Int. Price List)

**product\_function:** possíveis funções dos produtos (From Int. Price List)

**product\_handle:** possíveis formas de manuseio dos produtos (From Int. Price List)

**product\_raw\_material:** possíveis matérias primas dos produtos (From Int. Price List)

**product\_sell\_price:** preço de venda dos distribuidores por produto

**product\_suggested\_price:** preço de venda sugerido pela Delta (Promoções)

**product\_trasfer\_price:** preço de transferência dos distribuidores por produto

**product\_builder\_price:** tabela com os preços construtora dos produto

**product\_consumer\_price:** tabela com preços consumidor do produto

**product\_resale\_price:** tabela com preços revenda do produto

**product\_international\_list\_price:** tabela com preços internacional do produtos (From Int. Price List)

**product\_vrc:** possíveis estados da VRC dos produtos (From Int. Price List)

**representant\_budget:** todos orçamentos estarão nesta tabela, mesmo aqueles que não foram salvos

**representant\_budget\_payment\_condition:** possíveis formas de pagamento dos orçamentos

**representant\_budget\_product:** produtos do orçamento

**order:** sempre que um orçamento é processado suas informações são transmitidas para esta tabela

**order\_payment\_condition:** possíveis condições dos pedidos

**order\_status:** possíveis status de um pedido no sistema

**order\_product:** produtos do pedido

**order\_product\_note:** anotações do produto do pedido

**sub\_order:** sempre que um pedido é destinado suas informações são transmitidas

**sub\_order\_note:** anotações do subpedido

**sub\_order\_payment\_condition:** possíveis formas de pagamento do subpedido

**sub\_order\_product:** produtos do subpedido

**sub\_order\_status:** possíveis status do subpedido

**invoice:** sempre que itens de um sub\_pedido é faturado, sua fatura é gravada nesta tabela

**invoice\_item:** produtos referentes a fatura na tabela invoice

**invoice\_status:** possíveis status da fatura no sistema

**installment:** sempre que uma duplicata de uma fatura é processada ela é gravada nesta tabela



**installment\_status:** possíveis status de uma duplicata no sistema

**fee\_report:** tabela com todos relatórios de comissões gerados por usuários do sistema

**fee\_report\_item:** as duplicatas referentes ao relatório de comissões

**representant\_company:** tabelas com todas agências de representação parceiras da Delta

**representant\_company\_partner:** tabela com relação entre representações e representantes

**project:** tabela com projetos cadastrados no Project Tracker

**project\_stage:** possíveis estágios dos projetos no sistema

**project\_type:** possíveis tipos de projetos no sistema

**project\_closed\_status:** possíveis estados de fechamento de projetos

**project\_influencer:** relação entre influenciadores e projetos

**influencer:** tabela com influenciadores cadastrados do Project Tracker e do Sistema

**influencer\_type:** tabela com possíveis tipos de influenciadores

**post:** tabela com os emails trocados entre os inbox's

**post\_cc:** usuários em cópia no post

**post\_tag:** todas possíveis tags para posts no inbox

**post\_user\_status:** tabela que relaciona usuários e estados do posts

**notification:** tabela com as notificações geradas automaticamente pelo sistema

**notification\_subject:** possíveis assuntos das notificações

**notification\_status:** possíveis status das notificações

**sellout\_invoice:** faturas do Sellout usadas em relatórios

**sellout\_invoice\_nature:** possíveis naturezas das vendas das faturas do sellout

**sellout\_invoice\_product:** itens das faturas do sellout

**cloud\_directory:** diretórios da API Cloud Storage usada pelos usuários distribuidores

**cloud\_file:** arquivos da API Cloud Storage enviada pelos usuários

**address:** tabela com os endereços cadastrados no sistema

**state:** tabela com todos estados brasileiros

**city:** tabela com todas cidades brasileiras

**country:** tabela com todos países

## Linguagens de Programação e Frameworks

### PHP 5.6

A linguagem de programação PHP foi escolhida pelo seu amplo suporte fornecido pelos sites de hospedagem no Brasil, algo que não acontece com linguagens como Java, Ruby ou Python e o seu preço de desenvolvimento ser muito menor do que linguagens como C# e VB. Caso tenha problemas com a linguagem por favor não deixe de ler a documentação oficial da linguagem no site: <http://php.net/docs.php>.

### Laravel 5.1

A framework Laravel foi escolhida devida sua similaridade com a plataforma ASP.MVC da Microsoft, uma vez que eu já possuía grande experiência na ferramenta. Felizmente, a framework, apesar de um pouco lenta, provou-se capaz de lidar com o tamanho da aplicação. Caso tenha dúvidas sobre seu funcionamento, consulte sua documentação oficial (Somente em inglês) no site: <http://laravel.com/docs/5.1>.

### Blade Templates

A linguagem de templates para o PHP Blade foi escolhida devida sua simplicidade e ótima interação com a framework Laravel. Caso tenha dúvidas sobre seu uso, verifique os tutoriais do site da framework laravel: <http://laravel.com/docs/5.1/blade>.

### Eloquent ORM

O ORM Eloquent foi um dos principais motivos de ter escolhido a framework Laravel para esta aplicação então mantive como padrão na hora desenvolver a aplicação, assim poupando tempo para adaptar o sistema a outro ORM. Quaso tenha dúvidas quanto seu uso acesse o site: <http://laravel.com/docs/5.1/eloquent>.

### JavaScript + jQuery

No front-end decidi usar javascript puro com ajuda do jQuery devido sua facilidade de produção. A aplicação se beneficiaria de recursos mais avançados como Angular.JS ou React.JS mas por causa do tempo de desenvolvimento reduzido e a ausência de testes unitários não foi possível utilizá-las. Para documentação do Javascript eu recomendo o site da Mozilla Foundation: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. E para informações sobre a biblioteca jQuery visite a documentação oficial no site: <https://api.jquery.com/>.

## Bootstrap 3 + Bootstrap JS

O layout da aplicação foi feito usando o sistema de grids da biblioteca bootstrap e certos widgets como modais e alerts foram feitos usando a biblioteca javascript da mesma, assim diminuindo a complexidade e peso da aplicação. Para mais informações sobre ambas verifique a documentação: <http://getbootstrap.com/css/>.

## jQuery DataTables

Esta biblioteca foi de grande ajuda na hora de desenvolver as tabelas dinâmicas da aplicação, oferecendo várias funcionalidades de procura e ordenamento. Aconselho fortemente a leitura da documentação desta biblioteca em seu site oficial: <https://www.datatables.net/>.

## Arquitetura

### MVC + Middlewares

A arquitetura escolhida para desenvolvimento da aplicação foi a **Model View Controller** devido sua flexibilidade e grande escalabilidade. Apesar de causar repetimento e perda de eficiência em algumas partes do código. Para executar tarefas repetitivas, autenticar usuários e gerenciar sessões utilizei o padrão de projeto **Middleware** que permite um processamento de estágios dos requests antes destes chegarem as controladoras. Seu uso trouxe um ganho em eficiência e velocidade de desenvolvimento mas causou conflitos com o VPN. (Veja seção Vulnerabilidade e Cuidados)

**Model:** Foi criado um model para cada tabela no banco de dados, assim podemos navegar entre as relações da tabela e adicionar um level de abstração que ajuda no desenvolvimento de processos complexos.

**View:** não criei classes para views, para adicionar comportamento e dados utilizei os controllers enviando models para os templates Blade.

**Controllers:** As controladoras ficam na pasta `app/Http/Controllers` e são invocadas pelo mapeamento no arquivo `routes.php`.

**Middlewares:** Os intermediários estão na pasta `app/Http/Middlewares` e são mapeados no arquivo `routes` na propriedade `$middleware` da classe abstrata `Route`.

## Funcionamento

A aplicação foi desenvolvida para funcionar como um sistema integrado, onde diversos usuários, com diferentes níveis de acesso, são capazes de executar ações comuns e específicas ao seu tipo. Apesar de sua flexibilidade, a arquitetura MVC não é a ideal para esse tipo de aplicação, mas como os objetivos e necessidades que o sistema iria

atender não eram tão claros na fase inicial de desenvolvimento, utilizar outras arquiteturas poderiam engessar o sistema e acabar matando-o.

## Autenticação

A autenticação é feita por middlewares, onde existem dois caminhos por onde uma requisição pode percorrer, uma delas é a **Guest** e a outra a **Auth**. Quando uma requisição não possui os cookies de sessão ou as credenciais para início de sessão não batem, o usuário é redirecionado para o intermediário *Guest* e caso a url que deseja visitar esteja cadastrada no arquivo routes.php como uma área de livre acesso, para a área desejada. Agora, se a requisição possui uma sessão ativa no servidor ela será redirecionada para o intermediário *Auth*, que de a permitirá acessar qualquer área privada da aplicação.

## Sessões

As sessões dos usuários são temporárias, isto é, se o usuário permanecer inativo por muito tempo, sua sessão será deletada. Para acessar o usuário da sessão atual basta executar o metodo estático **user** do classe **Auth**.

## Models

Todas entidades registradas no banco de dados e com equivalente model possui as seguintes propriedades:

**created\_at:** data de registro da entidade no banco de dados,

**updated\_at:** ultima vez que a entidade foi tocada pelo metodo touch()

**active:** se a entidade é ativa (assim não perdemos nenhuma informação)

## URL's

As URL's foram projetadas para serem o mais simples possivel onde você consegue acessar detalhes de entidades através de suas chaves primárias. Por exemplo, considere as seguintes entidades:

|Customer:

|Order:

|Project:

|customer\_id = 10

|order\_id = 45

|project\_id=20

Suas urls seriam as seguintes:

Customer: <http://deltafaucetrep.com/portfolio/detail/10>

Order: <http://deltafaucetrep.com/order/detail/45>

Project: <http://deltafaucetrep.com/tracker/project/20/detail>

**Manutenção**

**Suporte do Desenvolvedor**

**Licença**

**Vulnerabilidades e Cuidados**