**Documentação do**

**Sistema de Controle de Pedidos ITSA**

**Versão 1.3.0**

**Autores: Jonas da Silva Azevedo**

**Revisor: Willian Freitas**

**2019**

**ÍNDICE DETALHADO**

1. OBJETIVO DO PROJETO 3

2. ARQUITETURA DO SISTEMA 4

3. MODELO ENTIDADE RELACIONAMENTO 6

4. INSTALAÇÃO NO SERVIDOR 7

4.1. Inserção de Empresas (Companies) 8

4.2. Inserção de Lojas (Stores) 8

4.3. Inserção de Produtos (Products) 9

4.4. Inserção de Código de Produtos (protuctcodes) 10

4.5. Inserção de Usuários (Users) 10

5. CONFIG.INI 12

6. CHANGELOG 13

7. EM CASO DE DÚVIDAS 16

1. **OBJETIVO DO PROJETO**

Esse projeto teve o intuito de informatizar e descentralizar a inclusão de informações de pedidos dos lanches produzidos e entregues pela empresa ITSA minimizando o tempo de emissão dos relatórios de produção assim como facilitando a criação dos relatórios de entrega disponibilizados para os entregadores da empresa.

Além disso, o fato de informatizar os processos da empresa possibilitou o acesso às informações em tempo real de quaisquer lugares desde que o usuário possua conexão à internet.

1. **ARQUITETURA DO SISTEMA**

Para atender às necessidades da empresa, foi desenvolvido o sistema presente utilizando-se a linguagem de programação PHP que é líder de utilização em ambiente Web. Contudo, nenhum framework específico foi utilizado, dada baixa complexidade das lógicas de negócio da empresa, utilizando-se uma estrutura pura e livre de especificações ou restrições.

Para gerenciamento de dependências, foi utilizado o *composer* que torna o trabalho muito mais fácil para o desenvolvedor. Porém, como já mencionado anteriormente, as lógicas de negócio da empresa são de baixa complexidade não sendo necessário, até o momento, a adição de quaisquer componentes extras. O *composer* foi utilizado para que, num futuro próximo, ajude na implementação de novas *features* que necessitem componentes ou bibliotecas externas e, além disso, utilizamos o *autoload* gerado por ele.

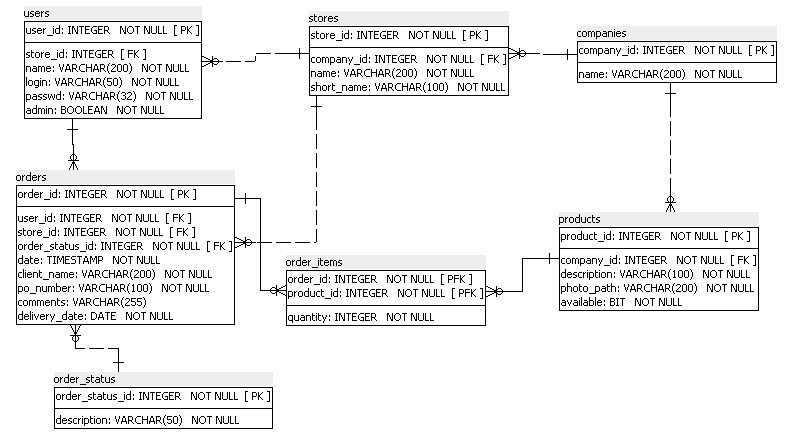
Seguem informações sobre a estrutura de pastas do projeto:

* */itsa/docs* – diretório onde se encontram todas as informações sobre o projeto como, por exemplo, esta documentação, arquivos do Modelo Entidade Relacionamento, etc.
* */itsa/resources* – esse diretório traz todos os recursos midiáticos utilizados pela aplicação, arquivos de estilo, plugins, arquivos javascript e imagens.
* */itsa/src* – diretório principal do sistema, aqui concentram-se as classes de modelo, as lógicas de negócio, os arquivos dos formulários do sistema e outros utilitários.
* */itsa/vendor* – pasta gerada automaticamente pelo *composer*. Não precisamos modificar nada dentro desse diretório.
* */itsa/DIST* – diretório onde consta o script SQL e a pasta dos arquivos do projeto para disponibilizar em ambiente de produção.

Quanto ao banco de dados, foi utilizado o MySQL que, se não o mais utilizado, é um dos maiores gerenciadores de banco de dados utilizados na comunidade principalmente pelo fator de ser gratuito e robusto.

1. **MODELO ENTIDADE RELACIONAMENTO**

Para suprir as necessidades do sistema, foi criada e aprovada a seguinte estrutura de dados:



O modelo apresentado acima foi desenvolvido pelo software *SQL Power Architect Community Edition* (<http://www.bestofbi.com/page/architect_download_os>) podendo ser baixado e instalado gratuitamente. O arquivo referente ao modelo acima encontra-se em */itsa/docs/itsa\_MER.architect*.

1. **INSTALAÇÃO NO SERVIDOR**

Para a instalação correta do sistema em qualquer servidor, certifique-se que o servidor Web instalado trata-se de um servidor HTTP Apache, pois os arquivos de configuração das rotas web do sistema (*.htaccess*) foram configuradas para esta distribuição em específico, podendo não ser compatíveis com outras distribuições de servidores HTTP.

É imprescindível que a pasta com os arquivos do sistema seja colocada na raiz do domínio! Por exemplo, se seu domínio é *www.meu-dominio.com*, a pasta do sistema deve ser colocada dentro da pasta raiz de acesso às páginas http do servidor, geralmente nomeadas *public\_html*, *www* ou *htdocs*, resultando no seguinte caminho para acesso à aplicação: *www.meu-dominio.com/itsa*.

Os arquivos necessários para todo o sistema encontram-se na pasta *DIST/itsa*, bastando apenas copiar a pasta *itsa* para o diretório raiz do servidor HTTP, como explicado acima. Feito isso, os arquivos do sistema estarão disponíveis para acesso, faltando apenas criar o banco de dados inicial do sistema. Lembre-se de alterar, se necessário, o usuário e senha de acesso ao banco de dados no arquivo *ConnectionManager.php* no diretório */itsa/DAO*.

Para a criação do banco de dados, basta executar o código do arquivo *DB.sql* que pode ser encontrado no mesmo diretório *DIST* onde também se encontra a pasta do projeto a ser instalado. Acesse o painel de controle do MySQL e copie todas as instruções SQL do arquivo e execute... a estrutura de dados padrão será criada apenas com a inclusão dos valores dos status dos pedidos (*Not Started, In Progress, Canceled, Completed*).

Execute também os arquivos .SQL encontrados na pasta DIST/Scripts.

Para a inserção dos outros dados, siga os subtópicos a seguir.

* 1. **Inserção de Empresas (*Companies*)**

Para inserir uma nova empresa ao sistema, basta ter o nome dela e executar o seguinte comando:

*INSERT INTO companies (name) VALUES (‘<nome\_da\_empresa>’);*

* 1. **Inserção de Lojas (Stores)**

De acordo com a estrutura de dados adotado nesse projeto, toda loja estará relacionada à uma empresa. Sabendo-se disso, para que seja possível inserir uma nova loja no sistema, o usuário deverá ter em mãos o nome da loja, um apelido para a loja e o ID da empresa cadastrada no sistema. Com essas informações em mãos, basta executar o seguinte comando:

*INSERT INTO stores (company\_id, store\_name, store\_short\_name, store\_code) VALUES (<id\_empresa>, ‘<nome\_da\_loja>’, ‘<apelido\_da\_loja>’, ‘<código externo da loja>’);*

O campo store\_code é usado como um campo de código externo da loja.

Ele é usado para identificar a loja no momento da importação do arquivo CSV com os produtos de uma ordem. Exemplo:

-> Nome do arquivo CSV a ser importado: bsx000146.csv

-> No momento da importação desse arquivo CSV, o sistema identificará a loja pelos três primeiros dígitos (neste caso: bsx).

-> A ordem irá ser associada a loja com store\_code igual à bsx.

* 1. **Inserção de Produtos (Products)**

A estrutura de dados aplicada neste sistema prevê que os produtos serão separados por empresas, ou seja, cada empresa terá uma gama de produtos disponíveis. Sabendo-se disso, é imprescindível obter o ID da empresa para se cadastrar um novo produto, além das informações do próprio produto que são descrição, caminho onde se encontra a foto do produto e se o produto consta como disponível para novos pedidos.

O diretório padrão para armazenamento das imagens dos produtos no sistema é /*itsa/resources/images/products*. Porém, no campo *photo\_path*, é necessário passar o caminho a partir do diretório *resources*. Por exemplo, o arquivo *bananas.jpg* do produto *Bananas* teria como campo *photo\_path* o seguinte valor: *resources/images/produtcts/bananas.jpg*.

Já o campo *available* trata-se do tipo booleano, ou seja, aceita valores 0 (falso/*false*) e 1 (verdadeiro/*true*), ou seja, para indicar que o produto está disponível basta inseri-lo com o valor 1 para o campo *available* ou 0 para o comportamento inverso.

Contudo, para se inserir um novo produto basta executar o seguinte comando:

*INSERT INTO products (company\_id, description, photo\_path, available) VALUES (<id\_empresa>, ‘<descrição\_produto>’, ‘<caminho\_imagem>’, <disponibilidade>);*

* 1. **Inserção de Código de Produtos (protuctcodes)**

A tabela de código de produtos é usada para associar os produtos com um código externo do mesmo. Ela possui dois campos, segue:

-> **product\_id:** chave estrangeira usada para identificar o produto (products.product\_id).

-> **product\_id\_external**: código externo do produto. Ele é usado para identificar o produto no momento da importação do arquivo CSV da referida ordem. Exemplo:

-> Uma linha do arquivo CSV contém a informação dos itens do produtos de uma ordem. Segue um exemplo abaixo:

Y1551,CHC01,HAM & CHEESE CROISSANT,1,3.15,15.75

-> A primeira informação (Y1551) refere-se ao produto. Este valor deve ser informado na coluna **product\_id\_external** para identificar o produto.

* 1. **Inserção de Usuários (Users)**

Os usuários deste sistema, são divididos em duas categorias: administradores e comuns. O que os diferenciam é a capacidade de cadastrar, consultar, alterar e remover pedidos de sua loja de vínculo ou das lojas de quaisquer empresas cadastradas no sistema. Para identificar se determinado usuário é administrador ou não, usamos o campo *admin* da tabela *users* que trata-se de um campo booleano aceitando 0 (falso/*false*) para identificar usuários comuns e 1 (verdadeiro/*true*) para identificar usuários administradores.

Todos os usuários do sistema possuem vínculo com uma loja, ou seja, para o cadastro de um novo usuário é imprescindível que se tenha o ID da loja que o usuário está vinculado. Além dessa informação, é necessário também ter o nome do usuário, um login e uma senha que deve ser criptografada em MD5. Para a criptografia da senha, recomendamos o site <https://www.md5hashgenerator.com/>.

Com as informações em mãos, execute o comando abaixo:

*INSERT INTO users (store\_id, name, login, passwd, admin) VALUES (<id\_loja>, ‘<nome\_usuario>’, ‘<login>’, ‘<senha\_md5>’, <administrador>);*

1. **CONFIG.INI**

O arquivo config.ini está na raíz do projeto.

Hoje ele é usado para identificar o diretório de onde serão lidos os arquivos .CSV para a importação da ordem.

Ele contém a seguinte estrutura:

**[csv]**

**directory\_read=../../../csv/**

Ao finalizar a importação do arquivo .CSV, e caso este arquivo for importado do diretório que consta na entrada “directory\_read” da seção “[csv]” do arquivo “config.ini”, o mesmo será deletado do referido diretório.

Obs: Nada impede de importar arquivos .CSV de outro diretório, mas o mesmo não será deletado do computador.

1. **CHANGELOG**

* *ITSA\_1.1.0*
  + Inclusão do relatório de produção por período
  + *Procedimentos para atualização*
    - Copiar os arquivos/pastas relacionados abaixo que podem ser encontrados no diretório DIST seguindo a mesma estrutura de diretórios no servidor:
      * itsa/.htaccess *(arquivo)*
      * itsa/resources/css/period-report.css *(arquivo)*
      * itsa/resources/plugins/momentJS (*pasta*)
      * itsa/resources/plugins/google-charts (*pasta*)
      * itsa/src/DAO/ProductDAO.php *(arquivo)*
      * itsa/src/logic/orders/period-report *(pasta)*
      * itsa/src/view/default/header.php *(arquivo)*
      * itsa/src/view/orders/period-report.php *(arquivo)*
      * itsa/src/Util/DateTimeUtils.php *(arquivo)*
* *ITSA 1.1.1*
  + Resolução do bug de datas do relatório de produção por período
  + *Procedimentos para atualização*
    - Copiar o arquivo itsa/src/logic/orders/period-report/filter.php que pode ser encontrado no diretório DIST seguindo a mesma estrutura de diretórios no servidor.
* *ITSA 1.2.0*
  + Implementação de imports de arquivos e links das páginas relativos, tornando possível a renomeação da pasta padrão do sistema e até mesmo utilizar o sistema na raiz do domínio.
  + *Procedimentos para atualização*
    - Como muitos arquivos foram atualizados, se sugere que toda a pasta itsa no diretório DIST seja copiada para o servidor de produção. Porém, caso desejar copiar apenas os arquivos alterados, copiá-los para o servidor na mesma estrutura que se encontram no diretório DIST:
      * itsa/index.php
      * itsa/resources/js/orders/complete.js
      * itsa/resources/js/orders/info.js
      * itsa/resources/js/orders/new.js
      * itsa/resources/js/orders/order-utils.js
      * itsa/resources/js/orders/previous-orders.js
      * itsa/resources/js/orders/today-orders.js
      * itsa/src/logic/orders/check-new-order-integrity.php
      * itsa/src/logic/orders/validate-new-order.php
      * itsa/src/logic/orders/validate-order-info.php
      * itsa/src/logic/user/check-authorization.php
      * itsa/src/logic/user/sign-in.php
      * itsa/src/view/default/header.php
      * itsa/src/view/orders/complete.php
      * itsa/src/view/orders/delivery-docket.php
      * itsa/src/view/orders/info.php
      * itsa/src/view/orders/new.php
      * itsa/src/view/orders/period-report.php
      * itsa/src/view/orders/production-list.php
* *ITSA 1.3.0*
  + Implementação para possibilitar a importação de order via arquivo .CSV
  + *Procedimentos para atualização*
    - Executar os arquivos .SQL encontrados na pasta DIST/Scripts.
    - Definir o código da loja que irá constar no arquivo .CSV (os três primeiros dígitos do nome do arquivo .CSV). A referida definição deve ocorrer no campo store\_code da tabela stores.
    - Definir no arquivo itsa/config.ini, na chave [directory\_read] da sessão [csv], o diretório de onde será importado os arquivos .CSV. Os mesmos devem estar em um diretório visível para o servidor. Exemplo: itsa/csv, onde que para este exemplo o valor da chave deve ficar: ../../../csv/
    - Copiar os arquivos/pastas relacionados abaixo que podem ser encontrados no diretório DIST seguindo a mesma estrutura de diretórios no servidor:
      * itsa/.htaccess *(arquivo)*
      * itsa/config.ini *(arquivo)*
      * itsa/resources/css/import\_csv.css *(arquivo)*
      * itsa/src/DAO/ProductDAO.php *(arquivo)*
      * itsa/src/logic/orders/import\_csv.php *(arquivo)*
      * itsa/src/logic/orders/import\_csv\_end.php *(arquivo)*
      * itsa/src/view/orders/new.php *(arquivo)*
      * Criar diretório (exemplo: itsa/csv) para os arquivos .CSV e configurar no arquivo itsa/config.ini

1. **EM CASO DE DÚVIDAS**

Caso haja a necessidade de tirar dúvidas sobre quaisquer informações relacionadas ao sistema, por favor, entre em contato com o desenvolvedor Alvaro Augusto Roberto pelo e-mail *alvaro.prog@gmail.com* ou mande um WhatsApp no número *+55 (19) 98922-8225*.