

Documentação - MT4 - Senha Segura

Autor: Mario de Moraes Barros Neto <mariombn@gmail.com>

Arquitetura do Sistema:

- **/mt4** - Pasta Raíz da Aplicação
 - **/config** - Pasta com o arquivo de configuração
 - **/migrations** - Pasta com o script de geração do Banco de Dados
 - **/public** - Pasta pública da Aplicação com os arquivos de Front-end
 - **/src** - Pasta com os códigos da Aplicação divididos da forma explicada abaixo
 - **/Entity** - Classes de Entidades (Não são Fabricadas)
 - **/Layout** - Arquivos de Layout para serem renderizados na View
 - **/Repository** - Classes responsáveis por todo tipo de persistência do Sistema
 - **/Service** - Classes de Serviço contém todas as regras de negócio da Aplicação
 - **.gitignore** - Arquivo para não versionamento no git
 - **bootstrap.php** - Arquivo global da aplicação, responsável por carregar as configurações e fazer o autoload das classes
 - **README.md** - Arquivo com as instruções de Instalação e Requisitos de Sistema

Para a arquitetura do sistema, escolhi trabalhar com os Design Patterns de Factory e Repository e uma estrutura de Action > Service > Repository para trazer um pouco mais de abstração no código.

Na prática o sistema vai trabalhar da seguinte forma:

- Usuário faz requisição para uma página
- Página requisitada chama o bootstrap.php para carregar as configurações e o autoload.
- A parte de cima da página carregada vai ser como um “controller” que vai carregar os services necessário a partir das suas respectivas fábricas
- A classe ViewService também pode ser fabricada para renderizar a tela com o template bootstrap,

Sobre as Classes e Métodos

Tarefa 1 - CRUD de Dispositivos

A classe `\Repository\DispositivoRepository` é responsável por todo o CRUD dos dispositivos com os métodos de Inserção, Alteração, Exclusão, Carregamento e Listagem.

Por se tratar de uma classe de Repositório, ela não tem nenhum tipo de regra a mais como tratamento ou validação das informações.

A classe `\Entity\DispositivoEntity` representa a entidade dos dispositivos com um atributo representando cada campo do Banco de Dados e um método getter e setter para cada atributo.

A classe `\Service\DispositivoService` possui todas as regras de negócios relacionadas ao CRUD de dispositivos mais os métodos responsáveis por disparar os comandos SSHs para os dispositivos cadastrados no sistema.

Para a parte de criptografia e Hashes, foi adotado uma única interface.

Para a criptografia, a interface obriga a classe a possuir dois métodos públicos, um para efetuar a criptografia e outra para descriptografar. Ambas recebem uma chave digitada pelo usuário e a descriptografia só pode ser feita com a mesma chave que foi usada para fazer a criptografia.

Para os Hashes, foi adotado um modelo bem parecido, com a diferença que só existe um método explícito pela interface que é o método `gerarHash`.

Todo o processo de desenvolvimento pode ser acompanhado pelos commits realizados no github no seguinte link: <https://github.com/mariombn/mt4/commits/master>