Uma imagem contendo objeto

Descrição gerada automaticamente

|  |
| --- |
| ENGIE Digital |
| ConectAgro |
| Especificações para Módulos Externos |



**Conteúdo**

[1. Objetivo do Documento 3](#_Toc21020150)

[2. Tecnologias utilizadas 3](#_Toc21020151)

[2.1. Linguagens de Programação 3](#_Toc21020152)

[2.2. Frameworks 3](#_Toc21020153)

[2.3. API 3](#_Toc21020154)

[2.4. Multi-Empresa 3](#_Toc21020155)

[2.5. Estrutura de Módulos 3](#_Toc21020156)

[3. Ambiente de Execução 4](#_Toc21020157)

[3.1. Servidor de Aplicação 4](#_Toc21020158)

[3.2. Bancos de Dados 4](#_Toc21020159)

[4. Segurança 4](#_Toc21020160)

[4.1. Usuários 4](#_Toc21020161)

[4.2. Grupos de Privilégios 4](#_Toc21020162)

[4.3. Senhas 5](#_Toc21020163)

[4.4. Transferência de Dados 5](#_Toc21020164)

[5. Indicações de tecnologias 5](#_Toc21020165)

[6. Gerenciamento do projeto 6](#_Toc21020166)

[7. Definição de entregue 6](#_Toc21020167)

[7.1. Códigos fontes 6](#_Toc21020168)

[7.2. Infraestrutura 7](#_Toc21020169)

[7.3. Projeto 7](#_Toc21020170)

[7.4. Design Visual 7](#_Toc21020171)

[7.5. Aplicação web 8](#_Toc21020172)

[7.6. Widgets 8](#_Toc21020173)

[7.7. Tempo de resposta 8](#_Toc21020174)

[7.8. Dúvidas e questionamentos 8](#_Toc21020175)

[8. Anexos 9](#_Toc21020176)

[8.1. Exemplos de JSONs 9](#_Toc21020177)

|  |  |  |  |
| --- | --- | --- | --- |
| Histórico de Revisões | | | |
| **Versão** | Data | Responsável | **Observações** |
| 0.1.0 | 28/08/2019 | Marco Antonio Abreu | Início da especificação |
| 0.2.0 | 24/09/2019 | Marco Antonio Abreu | Primeira revisão das especificações |
| 0.3.0 | 03/10/2019 | Marco Antonio ABREU | Incluída seção de Indicações de tecnologias |
| 0.3.1 | 03/10/2019 | Marco Antonio ABREU | Correções de textos |
| 0.3.2 | 10/10/2019 | Marco Antonio ABREU | Correções de textos |
| 0.3.3 | 05/11/2019 | Marcelo Valle Franco | Correções nos textos |

# Objetivo do Documento

Documentar os requisitos, recomendações técnicas e convenções a serem seguidas na construção de novos módulos para o Sistema Maestro, orientando os desenvolvedores sobre as melhores práticas para uma perfeita integração.

# Integração

## Informações Básicas

O **Sistema Maestro**, assim como seu nome descreve, é a célula central de comando que rege, assim como uma orquestra, um conjunto de funcionalidades geradas por um conjunto de subsistemas ou módulos desenvolvidos pela Equipe Técnica da Engie ou por parceiros; que captam, cadastram, editam e mostram os resultados e as informações que o usuário precisa.

O **Sistema Maestro** é uma aplicação Web, construído para funcionar na nuvem, sendo acessado através de navegadores de computadores desktop, apps de dispositivos móveis, além de softwares (“clientes”) desenvolvidos especificamente para esse propósito.

O **Sistema Maestro** tem a função de permitir o acesso protegido às suas funcionalidades básicas e aos seus módulos externos, que funcionam por meio de iframes, dentro da estrutura protegida. A comunicação e troca de dados entre o **Sistema Maestro** e seus módulos, é efetuada através de simples APIs que garantem a segurança e agilidade no processo.

Os módulos desenvolvidos por terceiros podem estar hospedados em diferentes servidores, podem ser desenvolvidos com diversas linguagens de programação e diversos bancos de dados homologados pela Equipe Técnica da Engie. Eles normalmente são acessados através de iframes na tela do sistema integrador.

A homogeneidade visual é garantida pela utilização de uma biblioteca de estilos e componentes em comum. As bibliotecas gráficas homologadas pela Equipe Técnica da Engie são extremamente abrangentes e simplificam a construção do front-end das aplicações.

# Tecnologias Utilizadas

## Linguagens de Programação

* A linguagem de programação recomendada é **PHP 7**, com **HTML 5**, **JavaScript** e **CSS 3**.
* Poderão ser utilizadas outras linguagens, no entanto, será requerido o aceite pela equipe técnica da Engie. É pré-requisito obrigatório que a tecnologia adotada seja capaz de ser integrada ao sistema principal.
* Os códigos fontes devem estar validados na ferramenta **SonarQube**, de acordo com a linguagem utilizada.

## Frameworks

* No caso de utilização de PHP, é recomendada a utilização de um framework como o **Laravel** para desenvolvimento no **back-end**.
* Para o **front-end**, é recomendado o framework **React**. Este framework facilita a migração para dispositivos móveis através de conversão com framework **React Native**.
* É pré-requisito a utilização da biblioteca (JavaScript + CSS) **Limitless** ou **Material UI** para o desenvolvimento do **front-end**, de modo a garantir uma **única identidade visual** entre o sistema e seus módulos.

## Bibliotecas Gráficas

No momento, duas bibliotecas gráficas são recomendadas pela Equipe Técnica da Engie. São elas: **Material UI** (<https://material-ui.com/>) e a **Limitless** (<http://demo.interface.club/limitless/>). As duas trazem um enorme conjunto de componentes (CSS + JavaScript) suficientes para a montagem do front-end do sistema.

## API

* A **API** (Application Programming Interface) utilizada será o modelo **REST** (Representational State Transfer) e/ou **JWT** (Json Web Token), dependendo da utilização.

## Multi-Empresa

* O sistema funcionará em ambiente multi-usuário e multi-empresa. Cada usuário terá acesso apenas às informações da sua empresa.

## Estrutura de Módulos

* **Os módulos do sistema serão carregados e exibidos (através de iframes) no sistema principal**. Não necessitarão ter quaisquer dependências entre si para funcionarem, Exceto para o caso de integração previamente definidas. Casos excepcionais poderão ser analisados junto à equipe técnica da Engie.
* Cada novo módulo agregado ao sistema deverá ser habilitado individualmente para a empresa/cliente solicitante. Esta habilitação pode ocasionar cobranças para a empresa cliente, conforme contrato entre as partes. A habilitação será executada em prazo previsto, por um administrador do sistema principal.

# Ambiente de Execução

## Servidor de Aplicação

* O sistema principal e seu banco de dado estará hospedado em ambiente virtual na Amazon Web Serves (AWS).
* Os módulos podem ser hospedados no mesmo servidor, em máquinas virtuais ou em outro local definido pelo desenvolvedor, com aprovação da equipe técnica da Engie.
* As requisições feitas às APIs deverão correr sobre HTTPS, utilizando certificado SSL válido.
* Os end-points das API, com exceção dos de Login, deverão estar aptas para receber informações codificadas no formato Json, sempre acompanhados por token de validação.

## Bancos de Dados

* O sistema principal utilizará banco de dados MySQL 5.7.
* Os módulos podem utilizar o banco de dados de sus escolha, mas dependerão de concordância com a equipe técnica da Engie. Deverão ser, preferencialmente, um RDS separado, ou um serviço executado no próprio servidor de hospedagem do módulo; ou ainda em outra EC2 dedicada para essa finalidade, caso a primeira opção não esteja disponível. O banco de dados deve ser, preferencialmente, livre do pagamento de licenças. São recomendados: MySQL, PostreSQL, MondoDB.
* As conexões com os bancos de dados devem ser realizadas através de certificado SSL ou equivalente.

# Segurança

## Usuários do Sistema

* O usuário **desenvolvedor** (da equipe técnica da Engie) será o responsável pelo cadastro inicial dos demais principais usuários. Ele cadastrará um ou mais **administradores**, que poderão por sua vez, cadastrar os outros usuários de menor privilégio e também realizar outras operações de administração do sistema, como as configurações etc..

## Grupos de Permissões

* Por convenção, o VSD utilizará o conceito de grupos de usuários com privilégios pré-estabelecidos, ou seja, cada usuário pertence à apenas um grupo e os membros deste grupo tem seus direitos pré-definidos no próprio código do sistema principal e dos módulos. Estes grupos serão os seguintes:

|  |  |
| --- | --- |
| **Grupo** | **Descrição** |
| Administradores | Podem realizar qualquer operação no sistema, exceto as configurações básicas (de sistema operacional), restritas ao usuário desenvolvedor. Este grupo não terá acesso a dados dos módulos, exceto logs e configurações. |
| Supervisores | Podem visualizar todas as informações da pertinentes à sua área, realizar algumas operações especiais, além de supervisionar as tarefas executadas por todos os **operadores**. |
| Operadores | Podem visualizar informações básicas e realizar as operações do dia-a-dia, tais como lançamentos, cadastros simples e outras operações de natureza operacional. |
| Visualizadores | Usuários que podem apenas observar a atividade no sistema dados, sem possibilidade para qualquer alteração nos mesmos. |

As permissões detalhadas dos grupos de usuários serão naturalmente mais bem definidas, conforme o desenvolvimento do sistema.

## Senhas

* Nenhuma senha de acesso de usuário será armazenada aberta (como digitada) nas bases de dados ou qualquer outro meio. No seu lugar, devem ser armazenados apenas os respectivos hashes (ou solução mais segura), tornando inviável a recuperação da senha original. O algoritmo de hash será definido na especificação das rotinas de segurança, normalmente Sha256, Sha512 ou outro algoritmo, com chave de pelo menos 256 bits.
* As senhas, sempre que possível, devem ser especificadas de acordo com padrões estabelecidos de segurança, ou seja, utilizando-se de combinações de caracteres de difícil dedução por pessoas não autorizadas. Normalmente essa combinação de caracteres significa utilizar conjuntos de letras maiúsculas e minúsculas, números e caracteres especiais, além de ter um comprimento mínimo de pelo menos 8 caracteres.

## Transferência de Dados

* Todas as conexões de transferência de dados, requisições às APIs e conexões externas deverão ser realizadas utilizando protocolos seguros, tais como HTTPS, SSH, SSL ou SFTP.

# Indicações de Tecnologias

É pré-requisito que os módulos desenvolvidos por terceiros utilizem as tecnologias recomendadas pela Engie.

|  |  |
| --- | --- |
| **Tecnologias** | **Produtos Aceitos** |
| Sistema Operacional | Linux |
| Linguagem de programação | PHP 7, Python, Asp.Net Core e Java |
| Banco de dados | MySQL/Maria DB, PostreSQL e MongoDB |
| Framework Back-end | Laravel |
| Framework Front-end | React, HTML, CSS. JS |
| Framework JavaScript/CSS | Limitless ou Material UI |

**Observação**: Por questões de facilidade de manutenção, é necessário que os processamentos existentes nos módulos sejam executados preferencialmente no back-end, utilizando as linguagens de programação sugeridas. Nunca nos bancos de dados (através de stored procedures). Não devem ser utilizadas bibliotecas não homologadas. Agendamentos devem ser realizados através da tecnologia CRON, ou outro mecanismo do sistema operacional.

Qualquer framework, linguagem ou biblioteca que não tenha sido mencionada nesse documento, deverá ser **previamente homologada** pela equipe técnica da Engie para que possa ser utilizada no desenvolvimento.

# Gerenciamento do Projeto

A metodologia ágil Kanban será utilizada para o gerenciamento do projeto e individualmente para cada um dos seus módulos.

# Definição de Entregue

Para que um módulo ou atividade de desenvolvimento seja considerada finalizada (entregue), a equipe responsável deverá disponibilizar todos os itens abaixo, onde for aplicável.

## Códigos fontes

* + Os códigos fonte deverão estar publicados no gerenciador de versão do projeto oficial da Engie (**GitHub**) e estar **totalmente disponíveis** para a equipe técnica da Engie, durante todo o processo e ao final do desenvolvimento.
  + Os códigos devem compilar/funcionar sem falhas, de acordo com as especificações para cada módulo, tarefa, linguagem e ambiente necessários.
  + Nenhum código armazenado no gerenciador de versão pode disponibilizar dados sensíveis (senhas de acesso, dados de clientes, chaves de criptografia etc.).
  + Os arquivos de código deverão estar comentados. Sempre contendo informações relevantes, o nome da empresa e dos seus desenvolvedores responsáveis.
  + Todas as classes deverão indicar sua função e aplicabilidade. Da mesma forma os métodos, as entradas que ele espera e as possíveis saídas. Parâmetros opcionais, seus tipos e valores padrão (default). Estruturas complexas dentro de métodos também deverão estar devidamente comentadas, da maneira que um desenvolvedor que conheça a linguagem seja capaz de entender para que serve e como funciona.
  + O resultado do desenvolvimento deverá sempre ser demonstrado no momento da entrega, via PC, web ou dispositivo móvel; salvo em exceções que não permitam tal demonstração. Nesse caso deverão ser previamente esclarecidas junto à equipe técnica da Engie, sempre com alguma comprovação de funcionamento, como um log. Nos casos onde a implementação não tenha resultado visual, como cálculos, medições, envio de comandos etc., deverá ser criado um recurso temporário para demonstração do funcionamento, chamada em linha de comando, geração de arquivo texto com o resultado ou outro meio que seja aceito pela equipe técnica da Engie.
  + Os códigos fontes devem estar validados na ferramenta SonarQube, de acordo com a linguagem utilizada e os requisitos de qualidade estabelecidos pela Engie.

## Infraestrutura

* + Havendo necessidade da criação de algum tipo de infraestrutura para a entrega do módulo ou parte do mesmo, deverão ser disponibilizadas as senhas de administração (root ou equivalente) para a equipe técnica da Engie.
  + Deverão ser criados diagramas descritivos das interligações entre as partes criadas, as previamente existentes e as externas ao módulo (caso existam). Exemplo: servidores web, roteadores, firewalls, load balancers, bancos de dados etc.
  + Deverão ser entregues, documentação referente aos requerimentos e especificações referentes aos módulos: suas configurações mínimas, máximas e recomendadas. Exemplo: versão do sistema operacional, da linguagem, servidor web, banco de dados, memória RAM, espaço estimado de armazenamento, processadores, velocidade, banda de rede/internet etc.

## Projeto

* + Sempre que necessário, o projeto do módulo ou artefato deve ser entregue tanto de forma textual como em forma de diagrama. Os diagramas necessários serão definidos para cada módulo ou artefato.
  + Banco de dados relacional: Diagrama Entidade-Relacionamento e comandos DDL para criação das tabelas, relacionamentos, constraints, triggers, funções, etc.
  + Banco de dados não relacional: Estrutura descritiva das listas e demais estruturas do BD.
  + Processo: UML – Diagrama de Estado, Diagrama de Sequência e outros diagramas aplicáveis.
  + Fluxo de informação: DFD, UML ou outro diagrama que melhor represente o fluxo em questão.

## Design Visual

* + HTML, CSS, JavaScript, Codificação de Cores no formato hexadecimal RGB (#RRGGBB); sempre seguindo a biblioteca utilizada.
  + Exemplos (prints) de telas nas variações PC, Notebook, Celular e Tablet, conforme especificado abaixo.
  + Na ausência desses, devem ser entregues wireframes com as sugestões dos respectivos layouts.
  + A resolução mínima a ser considerada para as telas: HD (1280 x 720 pixels)
  + Tamanho físico mínimo das telas (diagonal)
    1. PC: normalmente Full HD;
    2. Notebook: 13” (13 polegadas);
    3. Celular: 5” (5 polegadas);
    4. Tablet: 8” (8 polegadas).

## Aplicação Web

* + Deve ser compatível com pelo menos com os 3 (três) principais navegadores (browsers) do mercado, com os que possuem mais de 5% do mercado no brasil e no mundo (o que for maior), em pelo menos suas 2 (duas) últimas versões, na data da entrega da história/tarefa. Este critério pode ser mais ou menos restritivo dependendo das necessidades do cliente ou do projeto.

## Widgets

* Os Widgets estarão sempre atrelados a um módulo. Deverão ocupar a área disponibilizada para eles no sistema principal, obedecendo as especificações estabelecidas.
* A documentação do widget deve indicar quais parâmetros são necessários para que ele seja utilizado.
* Os widgets de informações dinâmicas deverão ser capazes de se atualizar em intervalos pré-determinados ou sempre que for necessário, sem que ele precise ser carregado novamente.
* Widgets e formulários de entrada de dados deverão estar protegidos contra SQL Injection, Cross-Site Request Forgery, Cross-Site Scripting, Manipulação de Cookies e outras atividades de captura indevida de dados ou sabotagem do sistema.

## Tempo de Resposta

* O tempo de resposta de telas deverão durar no máximo de 3 (três) segundos. Para os casos onde este tempo não tenha com ser atingido ou possa ultrapassar este valor, deverá ser utilizada alguma forma de interação com o usuário, tais como ampulheta, barra de progresso ou outro meio de informar que o processo está em execução e não travado. Também poderá utilizar mensagens de aviso ou notificação para quando o processamento for concluído ou outro meio qualquer acordado entre a equipe de desenvolvimento e a equipe técnica da Engie.

## Dúvidas e Questionamentos

* Quaisquer dúvidas e questões sobre a entrega de um módulo ou algum desenvolvimento parcial, deverão ser solucionadas ou respondidas exclusivamente pela equipe técnica da Engie. A equipe técnica da Engie decidirá se o módulo ou algum desenvolvimento parcial foi totalmente entregue, parcialmente entregue ou não entregue, podendo ou não ser requisitada sua conclusão ou reconstrução.
* A equipe da Engie definirá previamente, junto com a equipe de desenvolvimento, o que se espera de um módulo para que ele cumpra as expectativas para ser considerado entregue e finalizado.

# Anexos

## Exemplos de JSON

### JSON de troca de dados entre o sistema principal e o módulo (Exemplo de sintaxe):

{

"fromApp":true,

"action":"loginUserApp",

"payload":{

"email":"user@user.com",

"password":"123"

}

}