Histórico de Versões

Data	Versão	Descrição	Autor	Revisor	Aprovado por
31/03/2019	001	Iniciando o DAS	Anderson	Everton	Equipe
22/04/2019	002	Inserindo Demais Informações para Finalizar o DAS.	Everton	Anderson	Equipe

Índice

Documento de Arquitetura de Software

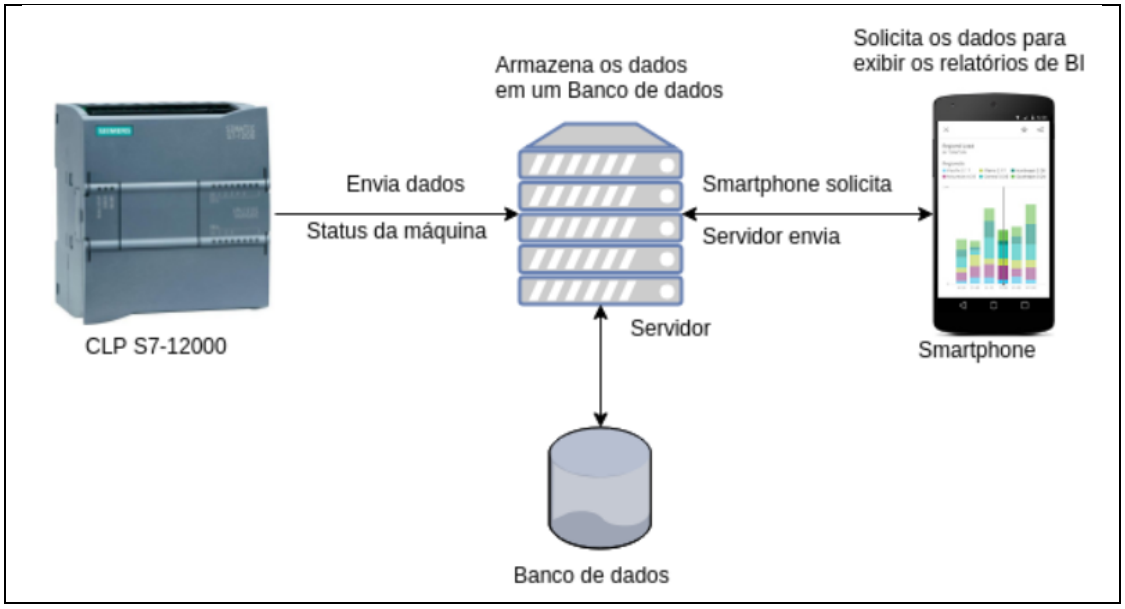
1. Objetivo do Documento

O Documento de Arquitetura de Software provê uma visão geral da arquitetura através de diferentes tipos de visões para descrever os diferentes aspectos do sistema.

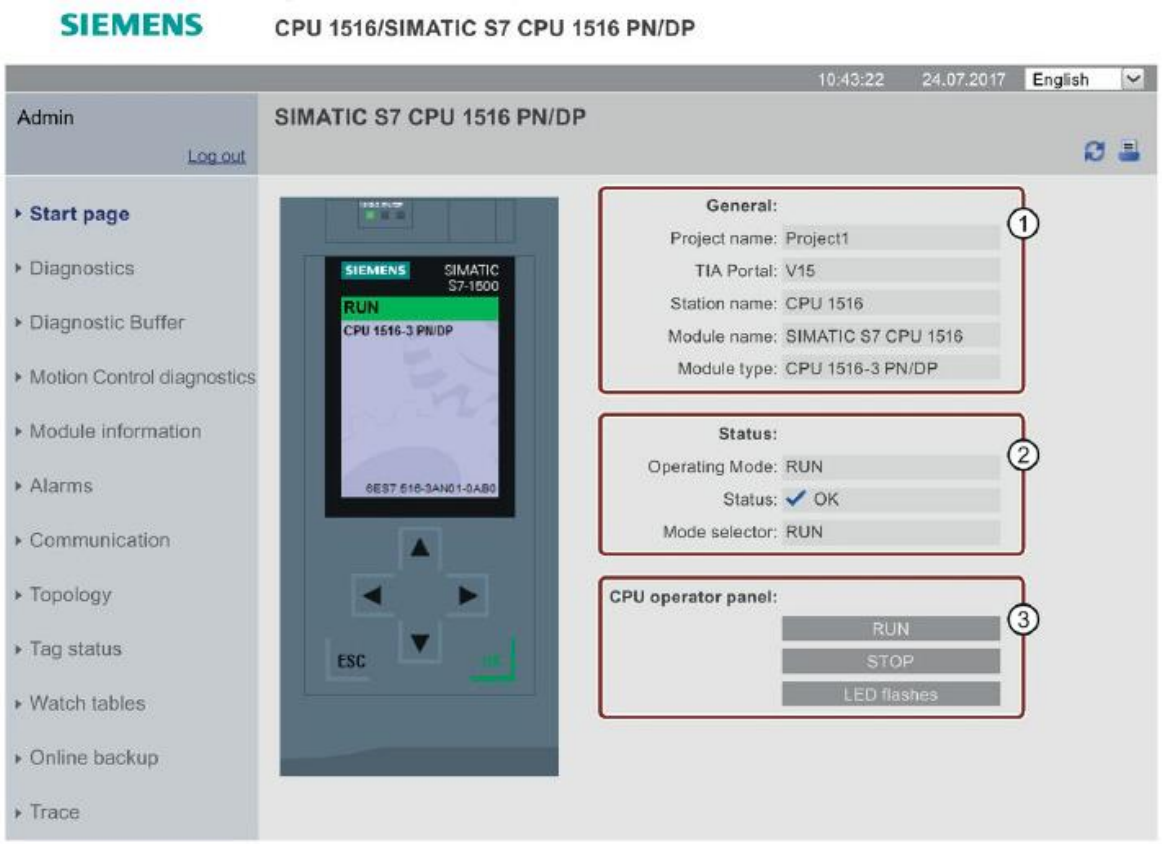
2. Objetivos e Restrições da Arquitetura

O projeto CODE Simatic foi criado devido a necessidade de se obter os dados gerados por uma máquina que possui o CLP S7-12000, salvá-los e gerar relatórios de BI que possam ser visualizados tanto em um aplicativo Android nativo ou híbrido.

Abaixo um diagrama que possui todos os hardwares do projeto, que tem o objetivo de relacionar todos os sistemas em um relatório de BI.



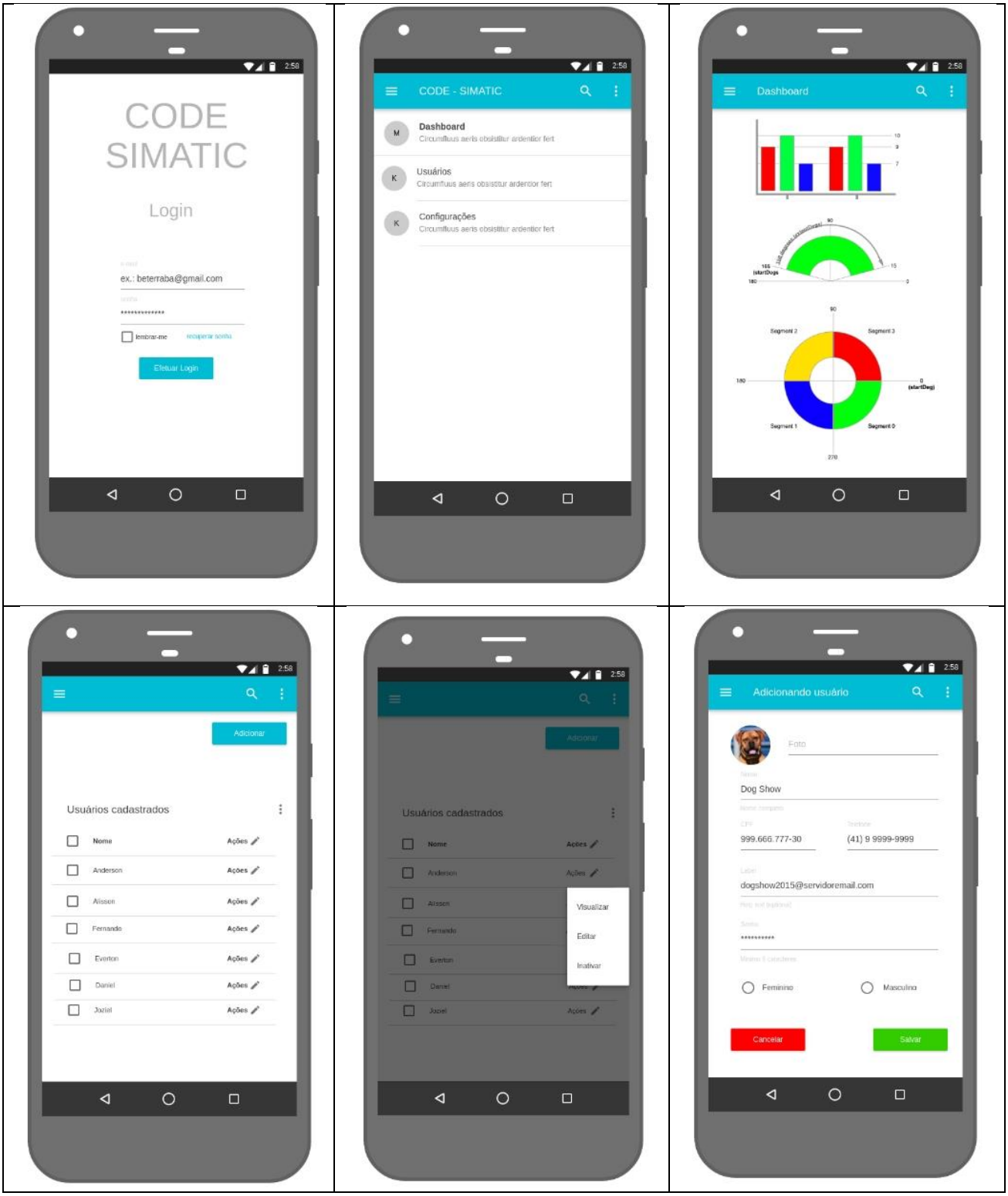
As restrições são decorrentes da máquina já ter um webservice próprio que exporta os dados em HTML dificultando a coleta e segmentação dos dados, conforme imagem a seguir.

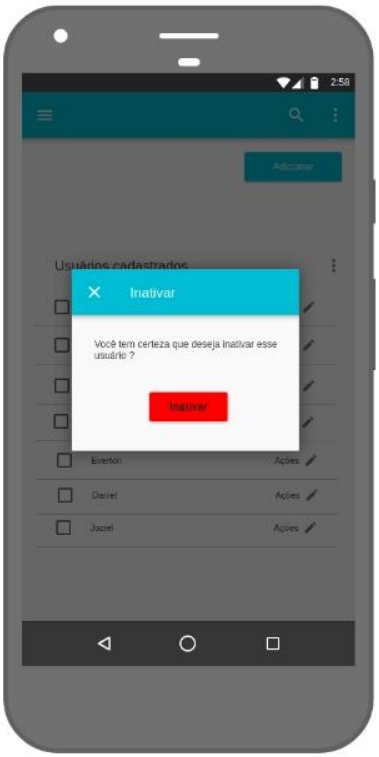


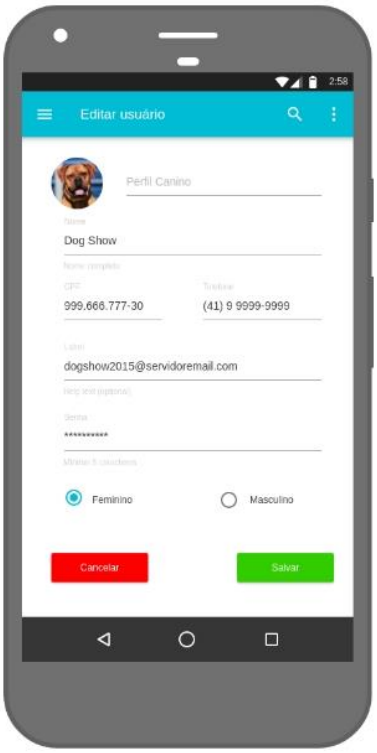
A máquina também tem momentos de manutenção e baixa de produção dificultando os testes na coleta de dados.

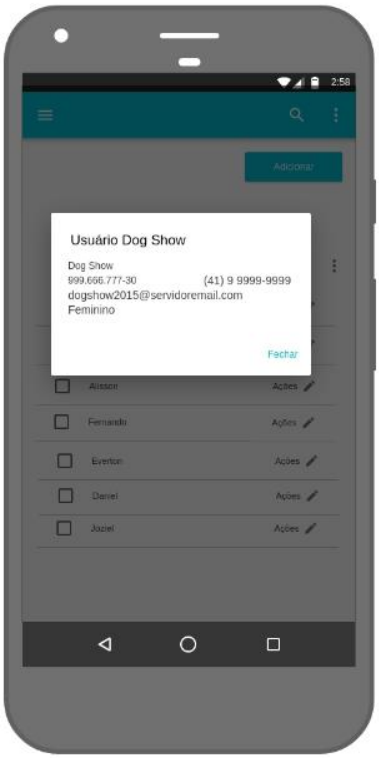
3. Elementos Arquiteturalmente Significativos

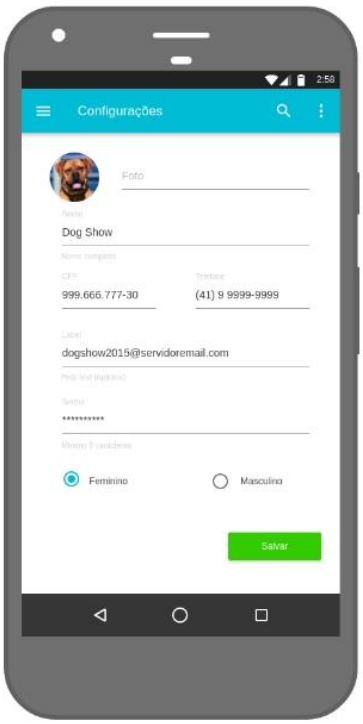
O componente de *Dashboard* mostrando de forma visual as informações geradas pela SIMATIC, buscando facilitar a compreensão.

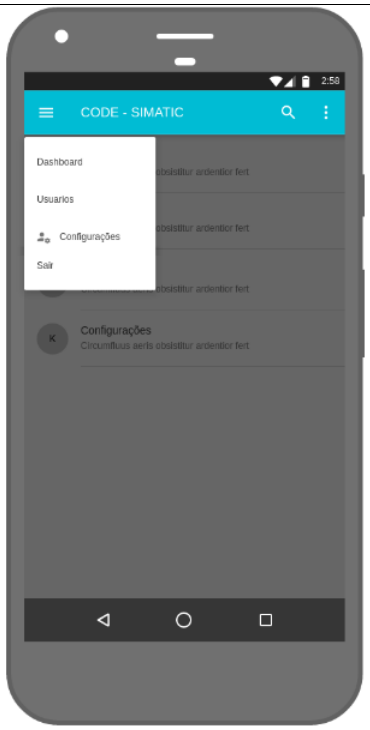


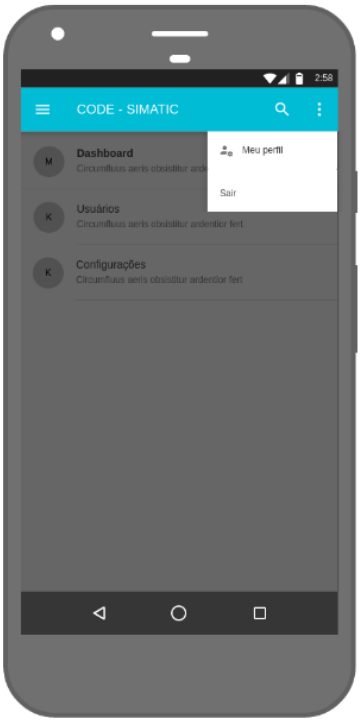

Inativar
Você tem certeza que deseja inativar esse usuário ?
Inativar


Editar usuário
Perfil Canino
Dog Show
999.666.777-30
(41) 9 9999-9999
dogshow2015@servidoremail.com
Feminino
Masculino
Cancelar
Salvar

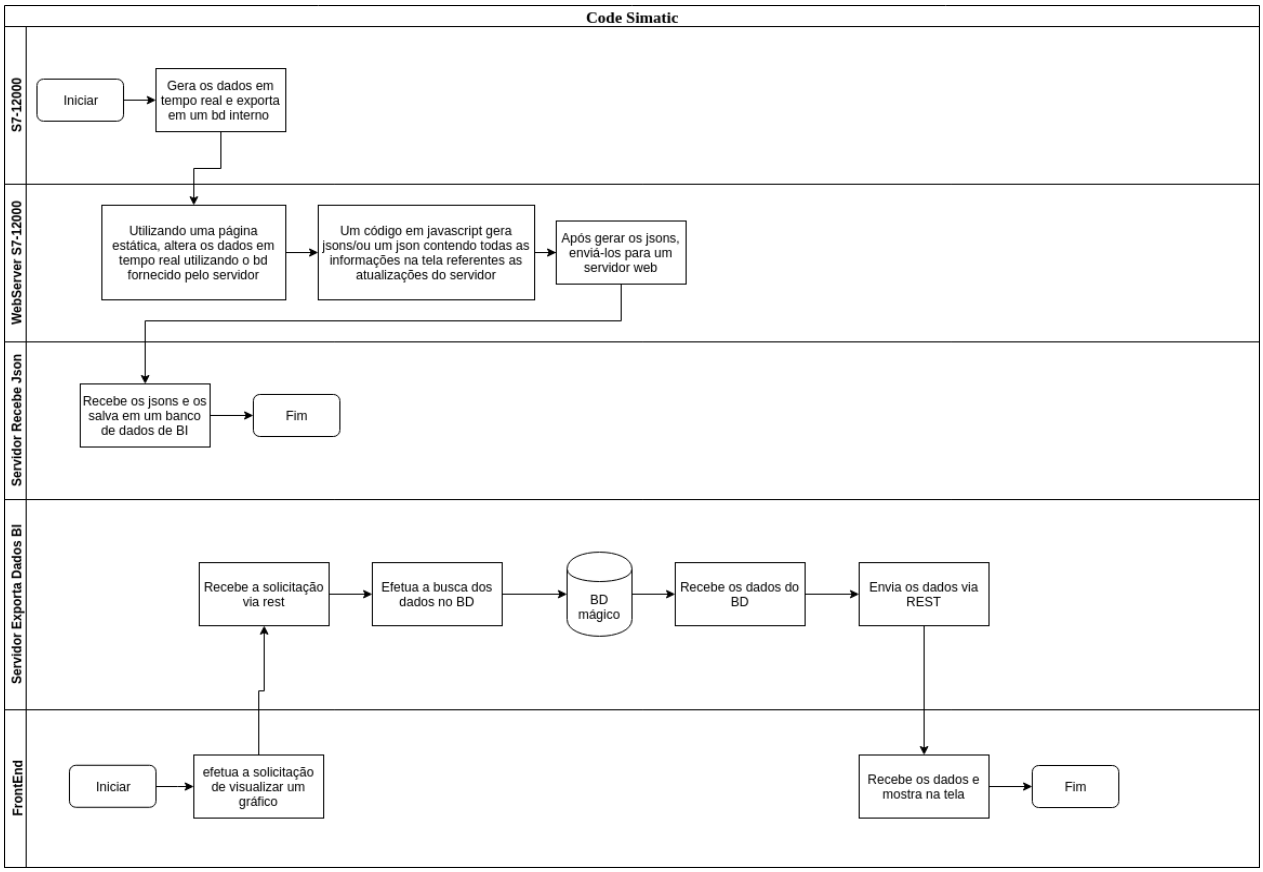

Usuário Dog Show
Dog Show
999.666.777-30
(41) 9 9999-9999
dogshow2015@servidoremail.com
Feminino
Fechar


Configurações
Foto
Dog Show
999.666.777-30
(41) 9 9999-9999
dogshow2015@servidoremail.com
Feminino
Masculino
Salvar


CODE - SIMATIC
Dashboard
Usuarios
Configurações
Sair
Configurações


CODE - SIMATIC
Dashboard
Usuários
Configurações
Meu perfil
Sair

4. Descrição da Arquitetura



4.1. Camadas e Subsistemas

Get usuário: Retorna um usuário.

url:http://code-simatic/rest/usuario/id

Entrada:
Id usuário
Saída:
Json usuário

Post login: Efetua login

url:http://code-simatic/rest/login

Entrada:
{
Email:<String>,
Senha:<String>
}
Saída:

Json contendo autenticação.

Get dados máquina: retorna os dados da máquina que serão convertidos em gráficos

url:http://code-simatic/rest/dados-maquina

Entrada:
Json contendo a data mínima e máxima dos dados a serem obtidos
Saída:
Json contendo os dados solicitados

Post usuário: insere um usuário no servidor

url:http://code-simatic/rest/usuario

Entrada:
Json usuário sem id
Saída:
OK 200 contendo o id do usuário criado.

Put usuário: atualiza um usuário no servidor

url:http://code-simatic/rest/usuario/id

Entrada:
Json usuário alterado
Saída:
OK 200 contendo o id do usuário alterado.

Delete usuário

url:http://code-simatic/rest/usuario/id

Entrada:
Somente o id
Saída:
OK 200 contendo informando que o usuário foi excluído

4.2. Padrões e Mecanismos arquiteturais

Estamos utilizando o padrão MVC (Model View e Control), onde as classes ficam separadas mais especificadamente em:

Model:

- Model;

View:

- Interface;
- Activity;

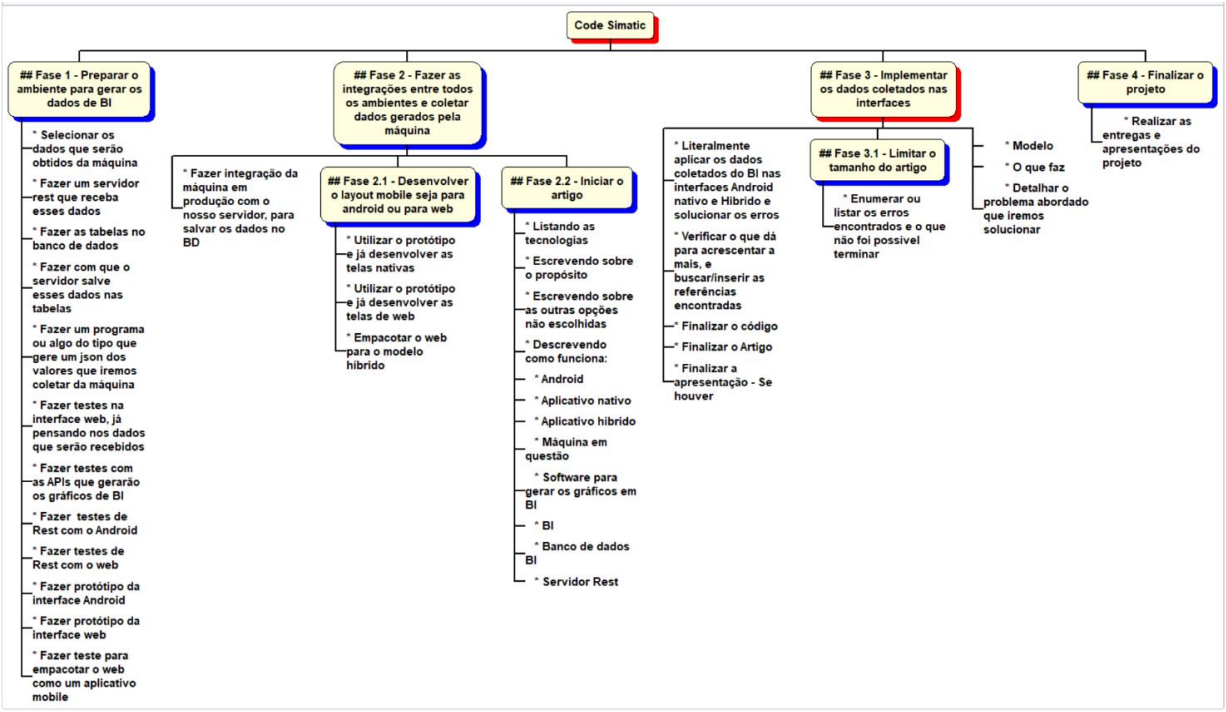
- Adapter;

Control:

- Dao;
- Rest;
- Útil.

4.3. Topologia

4.4. Visão geral da Implementação



5. Decisões e Justificativas

A equipe decidiu que precisava de modos para recuperar as informações do CLP e criar o método para gerar o JSON contendo essas informações. Corrigido HTML do servidor da máquina.

Buscamos informações da máquina Simatic, como manuais e documentação do WebService para entender seu funcionamento e parametrizar o formato de importação.

No entendimento da equipe teremos que subir os dados para o banco de dados e depois realizar a visualização no aplicativo Android com gráficos.