

Framework Utilizado: Flutter

Por que usar o flutter?

É um SDK de código aberto criado pelo Google para o desenvolvimento de aplicativos para dispositivos móveis. O flutter vem causando um grande entusiasmo na forma de desenvolver bons APPs para ambas plataformas (Android, IOS) utilizando o mesmo código transcrito em Dart.

Widget: Basicamente tudo em flutter é widget, desde uma screen completa a uma simples label. Como tudo se trata de uma árvore de Widget, sua abordagem acaba sendo interessante, pois facilita na compreensão e construção de views.

Existem Widgets que tem como papel organização e disposição de elementos como:

Container: Como diz o próprio nome: ele é somente um container para pôr um Widget, que nos possibilita setar parâmetros como padding, margin, background, etc;

Row: Organizam os widgets um do lado do outro;

Column: Organiza os widget um abaixo do outro;

Stack: Se comporta como uma pilha, adicionando um widget sobreposto ao outro.

Esses são os mais básicos e mais utilizados.

Além dos Widgets, existem os componentes que dão a cara para a aplicação como: Text, Icon, Image, RaisedButton, entre outros.

Montando um Layout:



Material: Esse Widget é um container pelo qual pode adicionar atributos de layout do material design como elevation, BorderRadius, etc.

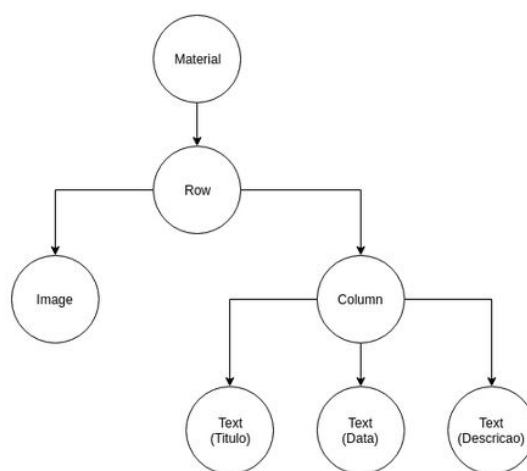
Row: Logo abaixo foi adicionado uma row para deixar a imagem e o Widget Column um do lado do outro.

Image: Widget utilizado para exibir uma thumb da notícia.

Column: Esse Widget nos possibilita deixar os textos (título, data e descrição) um abaixo do outro.

Text: Por fim o Widget que nos possibilita exibir um texto. Nele podemos configurar várias características em seu Style como textSize, textColor, entre outros. Foi utilizado para exibir o título, data e descrição da notícia.

Desenhando a árvore de Widgets do item da notícia temos esse resultado:



Linguagem Utilizada: Dart

Por que usar dart?

É uma linguagem de script voltada à web desenvolvida pela Google.

O Flutter é escrito em Dart, uma linguagem concisa, fortemente tipificada e orientada a objetos. O Dart é bem semelhante à linguagens como Swift, C#, Java e JS.

Introdução

Será feito um aplicativo para gerenciar uma horta inteligente, nele poderá ser feito um login e a partir do momento que você tiver acesso ao aplicativo, será possível visualizar uma versão da horta, sendo possível ver todos os dados dos sensores e até mesmo controlar o sistema de irrigação pelo aplicativo, como por exemplo ativar manualmente os irrigadores.

No desenvolvimento do aplicativo serão consumidos dados de uma API feitas por um dos grupos da matéria.

A proposta do projeto foi desenvolver uma aplicação mobile, onde fosse possível cadastrar os sensores existentes na horta e mostrar as informações coletadas pelos mesmos.

Objetivo Geral

Este aplicativo visa automatizar uma horta inteligente, recolhendo dados e oferecendo uma melhor forma de visualização dos mesmos pelo aplicativo, sendo possível verificar os dados de todos os canteiros e sensores ou até mesmo ativar os aspersores através do aplicativo.

Objetivos Específicos

- Visualizar dados dos canteiros;
- Visualizar dados dos sensores;
- Ativar os aspersores;
- Visualizar estação metereológica.

Diagramas

Diagrama de Caso de Uso

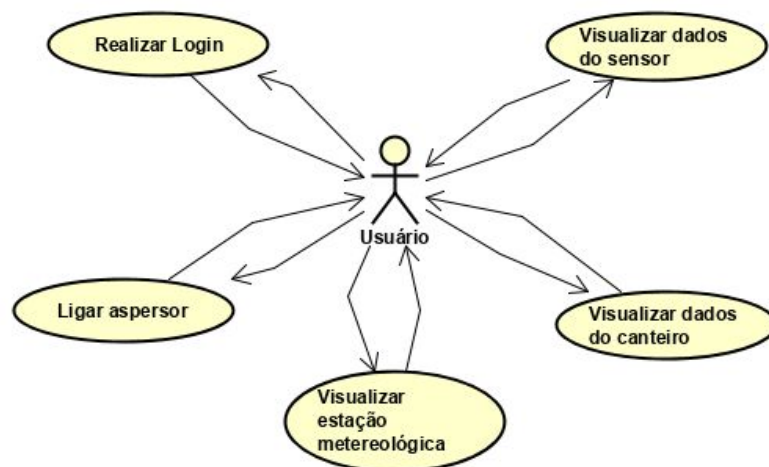


Diagrama de atividade

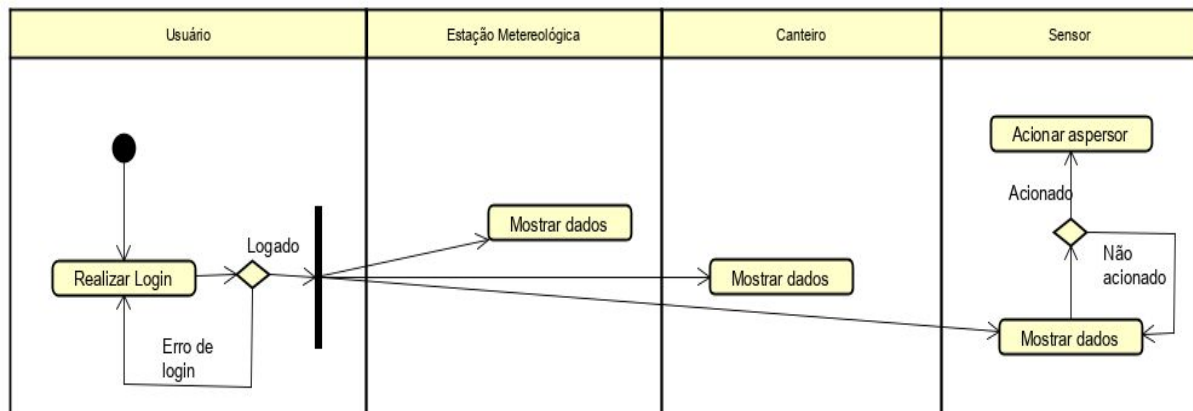
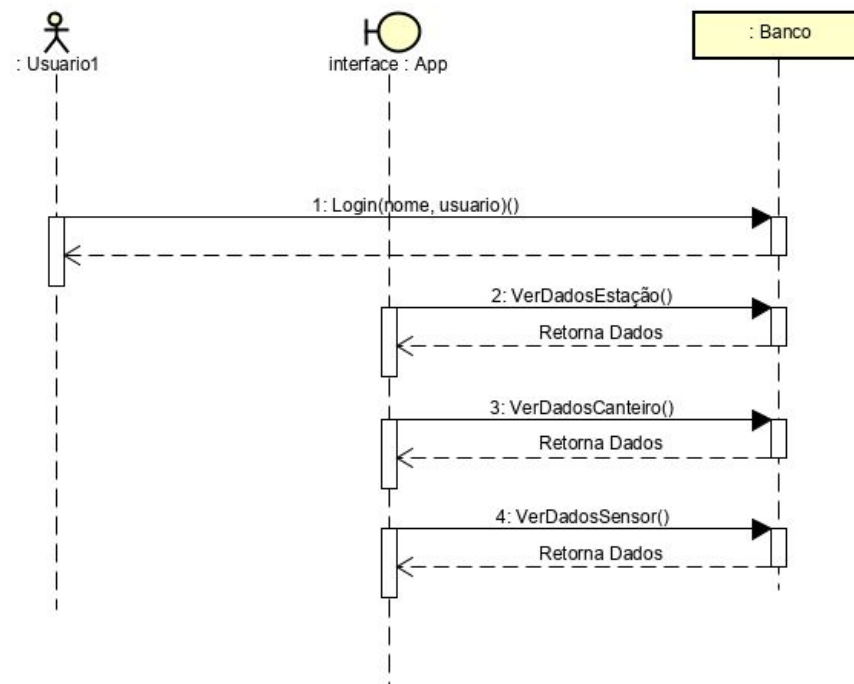


Diagrama de sequência

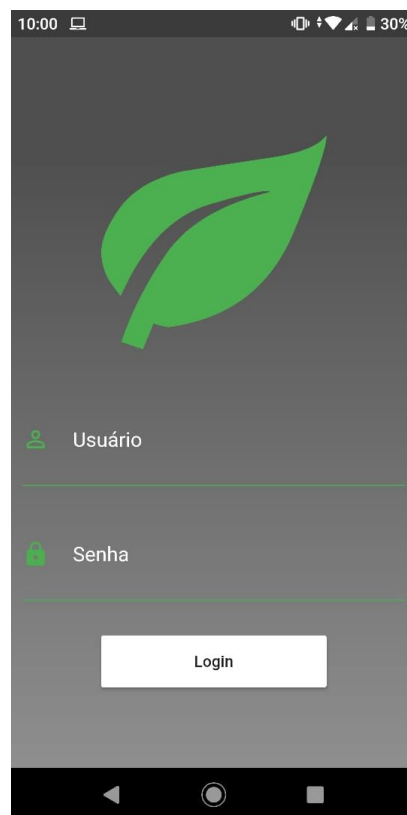


Telas do aplicativo

Login

Tela onde será feito o acesso ao aplicativo, realizando um login simples inserindo o nome de usuário no campo “Usuário” e a senha no campo “Senha”, para que o acesso seja liberado e o dispositivo possa utilizar as funcionalidades do aplicativo.

Os usuários permitidos para fazer login serão definidos diretamente no banco de dados, pois se fosse implementado uma tela para que a pessoa pudesse se cadastrar, qualquer pessoa poderia usar essa horta, no entanto o primeiro objetivo dela é apenas ser controlada por pessoas autorizadas. Seria um risco deixar qualquer pessoa utilizar pois elas poderiam mudar o estado dos aspersores, entre outras alterações que poderiam não ser desejadas.



Tela de Sensores

Tela onde após realizado o login, é possível visualizar os canteiros e os sensores, sendo feita a divisão por cada canteiro, e os sensores que fazem parte de cada canteiro, é possível visualizar na tela o nome dos canteiros, como “Canteiro 0”, etc., e seus respectivos sensores como “Sensor: 0”, etc. Além disso é possível visualizar a situação de um sensor, onde os “50%” na imagem representam a porcentagem de umidade em determinado sensor, é utilizado o “ON” para dizer que o sensor está ligado e “OFF” para dizer que está desligado.

As cores representam o estado da do sensor, sendo da seguinte forma:

Verde - Representa que o estado está satisfatório;

Laranja - O sensor precisa de algum tipo de atenção;

Vermelho - O sensor está apresentando algum problema que precisa ser resolvido.



Estação Metereológica

Nesta tela são apresentados os dados da estação meteorológica que serão captados e serão divididos entre alguns atributos previamente selecionados como:

Temperatura - Onde serão apresentados os dados em graus celsius da temperatura do ambiente;

Iluminação - Onde serão apresentados os dados da iluminação (em unidade de medida a ser definida);

Umidade do ar - Onde será apresentada a porcentagem de umidade identificada pelo sensor;

Altitude - Onde será apresentada a altitude identificada;

Pressão - Onde será apresentada a pressão da atmosfera(?) calculada pelo sensor.



Histórico dos Sensores

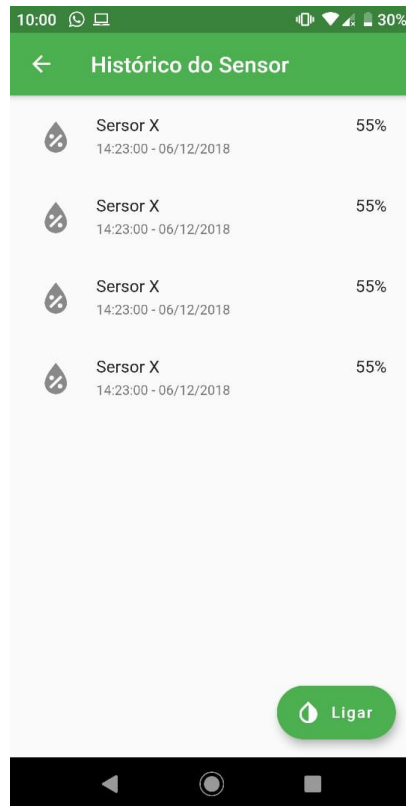
Esta é a tela que serão mostrados os históricos dos sensores, conforme dados apresentados por eles anteriormente e ela será dividida da seguinte forma:

Nome do sensor - Onde será apresentado o nome do sensor, para melhorar a identificação do sensor por exemplo "Sensor 1";

Data de atualização do sensor - Data no qual o sensor foi atualizado;

Umidade do Ar - Umidade do ar na data que o sensor foi atualizado;

Botão Ligar - Botão para ativar determinado sensor.



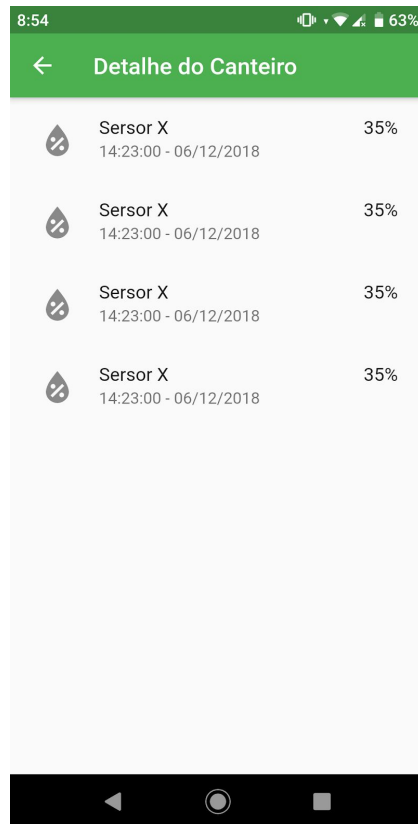
Detalhes do Canteiro

Esta é a tela que serão mostrados os detalhes dos canteiros, conforme dados apresentados por eles anteriormente e ela será dividida da seguinte forma:

Nome do sensor - Onde será apresentado o nome do sensor, para melhorar a identificação do sensor por exemplo "Sensor 1" em determinado canteiro;

Data de atualização do sensor - Data no qual o sensor foi atualizado;

Umidade do Ar - Umidade do ar na data que o sensor foi atualizado.



Integração com API

Medição:

Link: <http://horta.md.utfpr.edu.br/api/medicao>;

Get: Retorna os dados como id da lista, id do controlador, data que a medição foi feita.

Medição: Retorna uma lista com as medições:

tipo: Tipo de humidade;

valor: Valor da medição;

sensor: id do sensor.

Controlador:

Link: <http://horta.md.utfpr.edu.br/api/controlador>;

Get: Retorna os dados como id do controlador, mac do controlador, e a descrição que é uma string que junta o mac e o tipo do sensor.

Sensores:

código: Código do sensor;

sensor: Id do sensor;

id: id automático do node.

Sensor:

Link: <http://horta.md.utfpr.edu.br/api/sensor>;

Get: Retorna os dados de todos sensores como o id do sensor, o mac do controlador no qual o id pertence e o tipo por exemplo "ESP-8668".