

**UNIVERSIDADE PAULISTA – UNIP  
SISTEMAS DE INFORMAÇÃO**

Adriana Ferreira Moreira - D060CA2  
Beatriz Fialho Vaz - N925EF1  
Giovanna Rebeca Gerlach - C9494I2  
Laís Dias C. de Araújo - C976571

**ATIVIDADES PRÁTICAS SUPERVISIONADAS  
DESENVOLVIMENTO DE SISTEMAS DISTRIBUÍDOS  
SUSTENTABILIDADE – EM BUSCA DE UM MUNDO MELHOR**

**SÃO PAULO  
2019**

**UNIVERSIDADE PAULISTA – UNIP  
SISTEMAS DE INFORMAÇÃO**

**ATIVIDADES PRÁTICAS SUPERVISIONADAS  
DESENVOLVIMENTO DE SISTEMAS DISTRIBUÍDOS  
SUSTENTABILIDADE – EM BUSCA DE UM MUNDO MELHOR**

Projeto de Pesquisa apresentado ao Curso de Sistemas de Informação da Universidade Paulista, a ser utilizado como diretrizes para as atividades práticas supervisionadas.  
Orientador: Professor Lauro Rodrigues Tomiatti

**SÃO PAULO  
2019**

## Sumário

1.	OBJETIVO .....	4
2.	INTRODUÇÃO .....	5
3.	VISÃO GERAL: PROPOSTA .....	7
4.	O SISTEMA .....	13
5.	CONCEITOS BÁSICOS.....	14
5.1.	Web API .....	14
5.2.	JavaScript.....	14
5.3.	Angular JS .....	14
5.4.	Java.....	14
5.5.	Sprint Boot .....	14
6.	DESENVOLVIMENTO .....	15
6.1.	Prints.....	15
6.2.	Código.....	18
7.	CONCLUSÃO .....	54
8.	FICHA DE HORAS .....	55
9.	BIBLIOGRAFIA .....	57

## **1. Objetivo**

O presente trabalho tem como objetivo estudar os componentes que fazem parte de um sistema distribuído para que fique exemplificado como foi realizado o desenvolvimento de um website para exemplificar e demonstrar a importância de hábitos mais sustentáveis. A primeira parte do trabalho está exemplificando os motivos e necessidades de reciclagem, plantio, educação ao meio-ambiente, e o que o sistema precisa conter para demonstrar estes itens e deixar de uma forma interativa para que o usuário ao qual irá utilizar e navegar por esse Web Site esteja livre para contribuir com a manutenção deste. A segunda parte do trabalho será apresentado os conceitos de Web API, JavaScript, Angular, Java e SpringBoot e suas interações para que o trabalho de desenvolvimento seja realizado com sucesso.

Esta pesquisa e trabalho justifica-se pela necessidade de buscar uma necessidade de um cliente e desenvolver uma aplicação seguindo as diretrizes e o conteúdo que foi aprendido e na matéria de desenvolvimento de sistemas distribuídos.

## **2. Introdução**

Nosso trabalho foi baseado em pesquisas sobre o tema sustentabilidade, com o foco em reciclagem de lixo, reciclagem de óleo e também no plantio de árvores, trazendo principalmente um foco em árvores típicas brasileiras.

O primeiro item a falar é exatamente sobre estes termos, descrevendo sobre o que são, sua importância, como o fazer, a fim de deixar claro qual sua serventia e quais são os objetivos do sistema ao tratar sobre isto.

Conceituando, reciclar é reaproveitar produtos que iriam ser jogados fora e dar a eles uma nova funcionalidade.

A reciclagem é uma maneira de ajudar a preservar o meio ambiente. Ela colabora para a diminuição do volume de resíduos nos aterros sanitários, protegendo os solos e os lençóis freáticos (água pouco profunda); e transforma lixo em matéria-prima, aumentando a vida útil dos materiais e evitando que mais detritos sejam produzidos desnecessariamente.

O site ciclo vivo, listou os 5 motivos para realizar o plantio de árvores:

### **1. Aquecimento global**

Uma árvore pode durar aproximadamente 4800 anos e em apenas em um ano, inala em média 12 kg de CO<sub>2</sub> e exala oxigênio suficiente para uma família de quatro pessoas, durante 12 meses. O plantio de árvores é uma das principais recomendações de especialistas para combater o aquecimento global. Elas “sequestram” o CO<sub>2</sub> (dióxido de carbono) e refrescam a atmosfera.

### **2. Desertificação**

As árvores estabilizam o solo nas zonas áridas e podem evitar que o vento leve embora a camada superior com nutrientes, prevenindo a desertificação.

### **3. Água**

Regiões desmatadas não conseguem absorver nem 10% da água da chuva, enquanto uma árvore adulta pode absorver até 250 litros de água, evitando que enchentes ocorram. Além disso, as raízes reforçam os solos e as folhas dispersam o fluxo da água. Um ambiente florestado faz com que a chuva não caia torrencialmente, assim a água penetra no solo e abastece rios e córregos.

### **4. Comunidades**

As árvores e outras formas de vegetação protegem e dão força à vida comunitária. Elas fornecem produtos comerciais, alimento, fibras, resinas e frutos e garantem a vida de milhares de agricultores em suas comunidades.

### **5. Qualidade de vida**

Elas também garantem a descontaminação da atmosfera, nutrição para plantações e oferecem sombra. Considere que muitas cidades recebem indicações de “melhor lugar para se viver” devido à grande quantidade de ruas e locais arborizados.

O foco do trabalho não foi apenas em apresentar conceitos, mas também demonstrar como se faz, demonstrando quais são os benefícios gerados.

### **3. Visão Geral: Proposta**

A proposta inicial para o desenvolvimento deste trabalho é o foco em sustentabilidade, falando sobre hábitos que podem ajudar a desenvolver um mundo melhor.

A principal abordagem vai ser sobre reciclagem e plantio, abrangendo os temas descritos abaixo:

- **Descarte do óleo de cozinha:**

O óleo de cozinha é muito utilizado no dia a dia para preparo de alimentos, a Secretaria do Meio Ambiente alerta sobre o descarte incorreto do produto, que pode contaminar a água e causar outros danos ao meio ambiente. Dados da Sabesp informam que 1 litro de óleo pode prejudicar até 25 mil de litros de água.

Para evitar os efeitos desse descarte incorreto, o óleo de cozinha deve ser descartado em um recipiente com tampa (embalagem de vidro ou garrafa PET), para ser enviado à coleta seletiva, normalmente reaproveitado na produção de biodiesel e sabão.

Seguindo este conceito temos exemplos de alguns locais que fazem coleta deste material, para posteriormente descartar ou reaproveitar de forma correta:

**Pontos de coleta:**

Aclimação: Rua Muniz de Souza, 1119 – Aclimação.

Horário de coleta: 6h às 22h.

Telefone: (11) 3208-4042.

Anhanguera: Av. Fortunata Tadiello Natucci, 1000.

Horário de coleta: 6h às 18h

Telefone: (11) 3917-2406.

Barragem de Guarapiranga:

Rua Doutor Caetano Petraglia Sobrinho, acesso 43. Av. Atlântica – Jd. Guarapiranga.

Horário de coleta: 8h às 17h.

Telefone: (11) 5524-8403.

Carmo: Av. Afonso de Sampaio e Souza, 951 – Itaquera.

Horário de coleta: 5h30h às 18h/ 5h30 às 19h (horário de verão).

Telefone: (11) 2748-0010 / 2748-5001.

Cemucam: Rua Mesopotâmia – Jd. Passárgada – Cotia

Horário de coleta: 7h às 18h

Telefone: (11) 4702-2126/4702-8404.

Ou então, uma solução caseira, para reaproveitar este óleo:

Como transformar o óleo de cozinha em sabão?

Ingredientes necessários:

- Óleo de cozinha usado;
- Peneira fina;
- Recipiente de alumínio para ir ao fogo;
- Soda cáustica, corante e essência (a cada 6 litros de óleo utilize 500 gramas de soda cáustica);
- Forma.

Modo de preparo:

1. Passe o óleo pela peneira para retirar o excesso de impurezas;
2. Aqueça o óleo no recipiente de alumínio a 60°C;
3. Adicione a soda cáustica, corante e a essência e misture até obter uma pasta cremosa.

**IMPORTANTE:** este procedimento deve ser realizado com a proteção de luvas e máscara para o rosto, evitando o contato e inalação dos compostos químicos.

4. Despejar a pasta líquida nas formas, esperar endurecer e cortar os pedaços na forma de barras.

- **Coleta de lixo seletiva:**

A coleta seletiva existe para reduzir e melhorar o aproveitamento do lixo que produzimos. Atualmente, segundo pesquisa realizada em abril de 2019, as cidades da região metropolitana de São Paulo produzem 27 mil toneladas de lixo por dia, e em 39 municípios, os lixos são encaminhados para 13 aterros sanitários.

Pensando nisso, precisamos pensar em todas as maneiras de reduzir o lixo, reaproveitar tudo o que for possível, e só depois pensar em enviar materiais para reciclagem. Essa forma de atuação é conhecida como 3 Rs. Ainda assim, existem alguns materiais que não podem ser reciclados e os mais conhecidos são:

**Papéis não recicláveis:** adesivos, etiquetas, fita crepe, papel carbono, fotografias, papel toalha, papel higiênico, papéis e guardanapos engordurados, papéis metalizados;

**Metais não recicláveis:** cliques, grampos, esponjas de aço, latas de tintas e pilhas;



**Plásticos não recicláveis:** cabos de panela, tomadas, isopor, adesivos e teclados de computador;

**Vidros não recicláveis:** espelhos, cristal, ampolas de medicamentos e lâmpadas.

A coleta seletiva de lixo é extremamente importante para a sociedade. Além de ser uma grande vantagem para o meio ambiente, reduzindo a poluição de rios e solos, gera renda para milhões de pessoas e economia para empresas. Assim possibilitamos o maior aproveitamento dos resíduos, que antes eram descartados e como não eram reaproveitados aumentavam a quantidade de lixo causando grandes problemas ambientais.

Em São Paulo, podemos contar com a Coleta Domiciliar Seletiva, presente em 94 distritos, onde as concessionárias Loga e Ecorubis realizam o recolhimento dos resíduos recicláveis e destinam prioritariamente para 25 Cooperativas de reciclagem, que são habilitadas na Amlurb, e depois para as Centrais Mecanizadas de Triagem. Ao todo, são coletadas cerca de 280 toneladas de recicláveis diariamente, uma média de 7 mil por mês, desse valor, 100% é direcionado às cooperativas. Assim que coletados, os resíduos passam por um processo de triagem para identificar os materiais recicláveis.

**Pontos de coleta:**

ENDEREÇO	TELEFONE	MATERIAIS	FORMA DE TRABALHO
R. Contos Gauchescos, 86 V. STA. CATARINA	5562-4276	Papel, pet e longavida	Retira sob consulta
R. Lofgren, 2109 – Vila Clementino	(11) 5080-7000	Brinquedos, moveis, alimentos, roupas (bazar)	Retira dependendo da quantidade, acima de quatro caixas grandes.
Rua São Gabriel, 428 – Vila Galvão – Guarulhos – faleconosco@andreluiz.org.br	2457-7733	Aceitam o material mas tem que estar com um a mais, como brinquedos, moveis, roupas e eletrodomésticos, pois eles mandam um caminhão e precisa ser muita coisa.	Retiram
Avenida Paulista, 2073 – 2º sub-solo Jardim Paulista.	3179-0038	Papelão, vidro, lata, alumínio, papel branco	Não retiram

Rua Galeno de Almeida, 659. PINHEIROS, CEP: 05410-030.	3064-3976	TODOS	Retira acima de uma ton. Nas proximidades
R. 25 de Janeiro 274, LUZ – CEP 01103-000	3311-9928	Papel, papelão, alumínio, plástico, vidro, metal	Não retiram
Rua Cinira Polônio, 369 São Rafael.	2752-9446	Todos, menos lâmpada e pilha	Retiram em locais próximos à cooperativa
Rua Padre Ostaqueo, 1700 – Poá.	4639-4028	TODOS	Retiram
Estr. Manoel Lages do Chão, 590 – Lajeado, Cotia – SP, 06705-050	4243-1810	Papelão, papel, vidro, lata, alumínio, metal, plástico	Retira em quantidade de uma perua, em bairros próximos
Avenida Francisco Matarazzo, 455	3865-4130	Papeis em geral, plásticos, vidros e metais	Não retira
Rua Joviano Pacheco de Aguirre, 255 – CAMPO LIMPO	5841-3366 / 95100-4202 (pode mandar foto do material por whatsapp pra mais infos)	Papel, plástico, metal, vidro, cartuchos	Retiram em toda São Paulo, dão prioridade a maiores quantidades
Ariston Azevedo, 10 – Belenzinho, São Paulo – SP, 03021-020	2509-6358 (Juliana)	Como doação: alumínio, plásticos, papel, papelão, revistas, embalagens Tetrapak, garrafas, potes e embalagens de vidro, chapas de RAIIX, EAD (de produto de limpeza). Compram: PET, Alumínio e jornal. Apenas isopor que não.	Retira Sob consulta
R. Junqueira Freire, 176 – Liberdade, São Paulo –	3209-4112	TODOS; incluindo raio x, bateria, vhs	Não retira

SP, 01507-020			
Diversos endereços. <a href="https://content.paodeacucar.com/sustentabilidade/es-tacoes-de-reciclagem-pao-de-acucar">https://content.paodeacucar.com/sustentabilidade/es-tacoes-de-reciclagem-pao-de-acucar</a> (lojas que aceitam no link acima).	3055-6767(SP) / 0800-7-732-732	Todos	Não retira
De SP e outras localidades	“ <a href="http://www.capital.sp.gov.br/cidadao/rua-e-bairro">http://www.capital.sp.gov.br/cidadao/rua-e-bairro</a> <a href="http://cempre.org.br/servico/cooperativas">http://cempre.org.br/servico/cooperativas</a> ”	Todos	Retira Sob consulta

- **Plantio de Árvore – árvores típicas brasileiras:**

As árvores são indispensáveis na natureza por possuírem diversas finalidades. Elas auxiliam na purificação e umidade do ar, pois agem como sequestradoras de CO<sub>2</sub>, capturando gases tóxicos e devolvendo oxigênio para a atmosfera. Além disso, as árvores trazem outros benefícios para vários ecossistemas e espaços públicos.

Quando uma área é desmatada pela atividade agrícola, por exemplo, o solo fica exposto às chuvas e faz com que as erosões e deslizamentos ocorram com maior frequência. Para evitar estes problemas, o plantio de árvores é necessário, pois os arbustos conseguem armazenar boa parte da demanda pluvial. Além disso, as árvores servem para sugar esta quantidade de água e abastecer os lençóis freáticos – camada subterrânea abaixo do solo.

Pensando nisso, abaixo trouxemos uma lista de algumas das árvores típicas brasileiras:

Nome	Nome Científico
Pata de Vaca	Bauhinia variegata
Quaresmeira	Tibouchina granulosa
Jacarandá de Minas	Jacaranda Cuspidifolia
Sibipiruna	Caesalpinia Peltophoroides
Manacá da Serra	Tibouchina mutabilis
Chuva de Ouro	Lophanthera lactescens
Ipê-branco	Tabebuia roseo-alba
Babosa Branca	Cordia superba
Ipê-amarelo	Tabebuia ochracea
Aroeira-vermelha	Astronium fraxinifolium
Angico-branco	Anadenanthera colubrina
Cambucá	Plinia edulis
Canela-louro	Nectandra megapotamica

Guabiroba-árvore	Campomanesia xanthocarpa
Grumixama	Eugenia brasiliensis
Açoita-cavalo miúdo	Luehea divaricata
Sapuva	Machaerium stipitatum
Jacarandá-paulista	Machaerium villosum

#### 4. O sistema

O sistema funcionará da seguinte forma:

O usuário poderá navegar pelo site para encontrar locais de que realizam a coleta de óleo, coleta seletiva de lixo e ainda visualizar uma lista com os diversos tipos de árvores nativas brasileiras. Como intuito de demonstrar a abrangência de nossa natureza e também a importância de sempre buscar alternativas para ajudar o meio ambiente, construindo assim, um planeta mais sustentável.

Nas páginas de navegação, serão executadas algumas ações. Na coleta de óleo e coleta seletiva de lixo, além de trazer a lista com os locais que realizam esse tipo de ação, terão as opções de inclusão, edição e exclusão. Para que assim o usuário consiga se sentir parte de uma causa maior, e entender que a inclusão de um novo local, ou a exclusão por não estar mais em operação, pode ajudar outras pessoas que estão em buscar destes.

As tecnologias utilizadas para este desenvolvimento serão:

Front-End: HTML e CSS.

Back-End: Spring Boot (Java).

As definições de conteúdo para o Back-End estão separadas em Web APIs. De maneira listada a baixo:

- ➔ Cadastro – Local coleta de óleo (POST);
- ➔ Cadastro – Local coleta seletiva de lixo (POST);
- ➔ Alterar – Local coleta de óleo (PUT);
- ➔ Alterar – Local coleta seletiva de lixo (PUT);
- ➔ Deletar – Local coleta de óleo (DELETE);
- ➔ Deletar – Local coleta seletiva de lixo (DELETE);
- ➔ Buscar – Local coleta de óleo (GET);
- ➔ Buscar – Local coleta seletiva de lixo (GET);
- ➔ Buscar – Árvores típicas brasileiras.

A integração entre o Front-End e o Back-End será realizada em AngularJS (JavaScript).

## **5. Conceitos Básicos**

### **5.1. Web API**

A sigla API vem do acrônimo em inglês que significa, Application Programming Interface, e traduzindo para o português Interface de Programação de Aplicações. Esta pode ser definida como um conjunto de rotinas e padrões estabelecidos e documentados por uma aplicação X, para que outras aplicações consigam utilizar as funcionalidades desta aplicação X, sem precisar conhecer detalhes da implementação do software. As APIs permitem interoperabilidade, ou seja, comunicação entre aplicações e usuários.

### **5.2. JavaScript**

JavaScript é uma linguagem de programação criada em 1995 por Brendan Eich. Tem como propósito de oferecer aos desenvolvedores formas de tornar determinados processos de páginas web mais dinâmicos, tornando seu uso mais agradável, permitindo controlar mídias, imagens animadas e outros. A linguagem é uma das tecnologias mais importantes e utilizadas na internet.

### **5.3. Angular JS**

O AngularJS é um framework (extensão do JavaScript) de desenvolvimento front-end para aplicações web, tem como foco simplificar a codificação e o processo de teste. Também permite os desenvolvedores web fazer uso do HTML para definir associações de dados, validações, e utilizar manipuladores de resposta para lidar com as ações do usuário de maneira que contribui para aceleração.

### **5.4. Java**

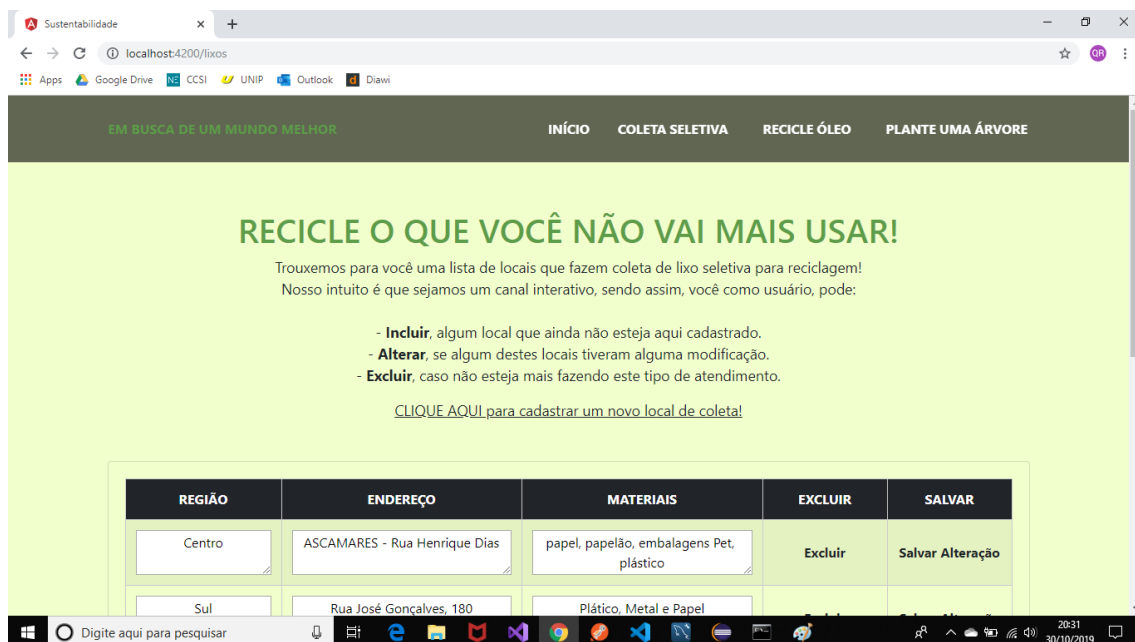
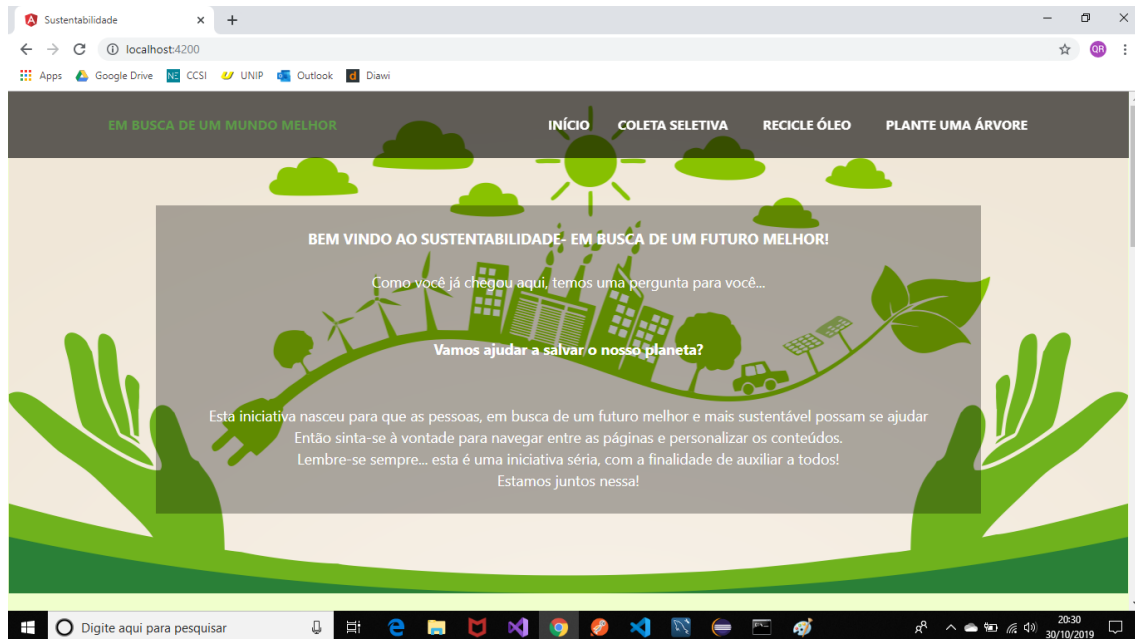
O Java é uma linguagem de programação e também plataforma computacional, rápido, seguro e confiável, que foi lançada pela primeira vez em 1995. Pode ser encontrado em console, computadores, datacenters, celular, internet e outros.

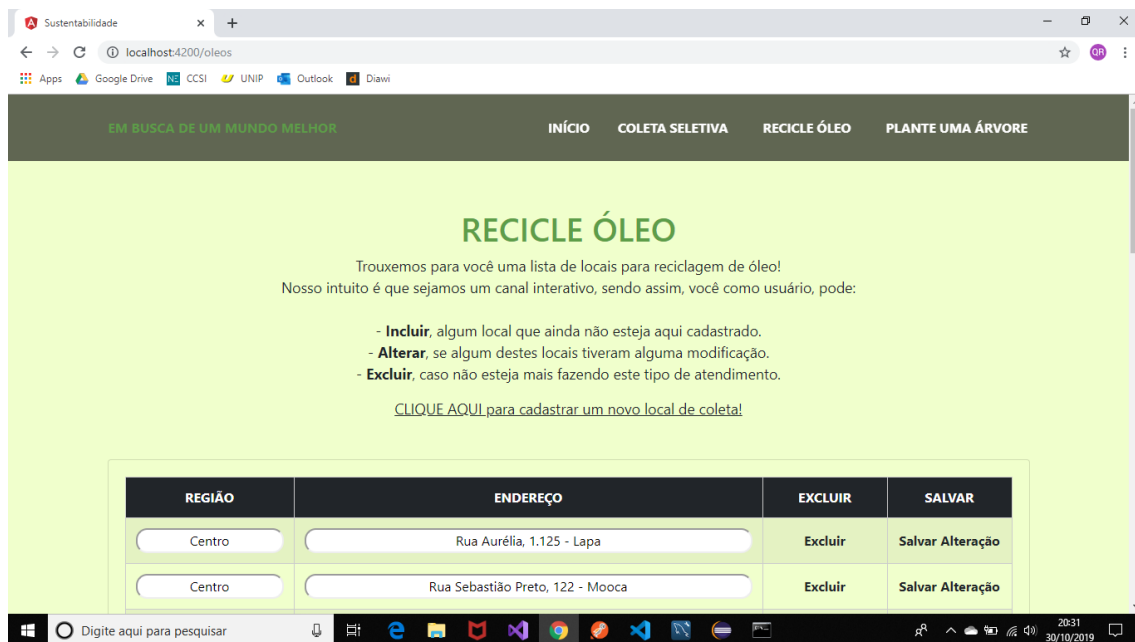
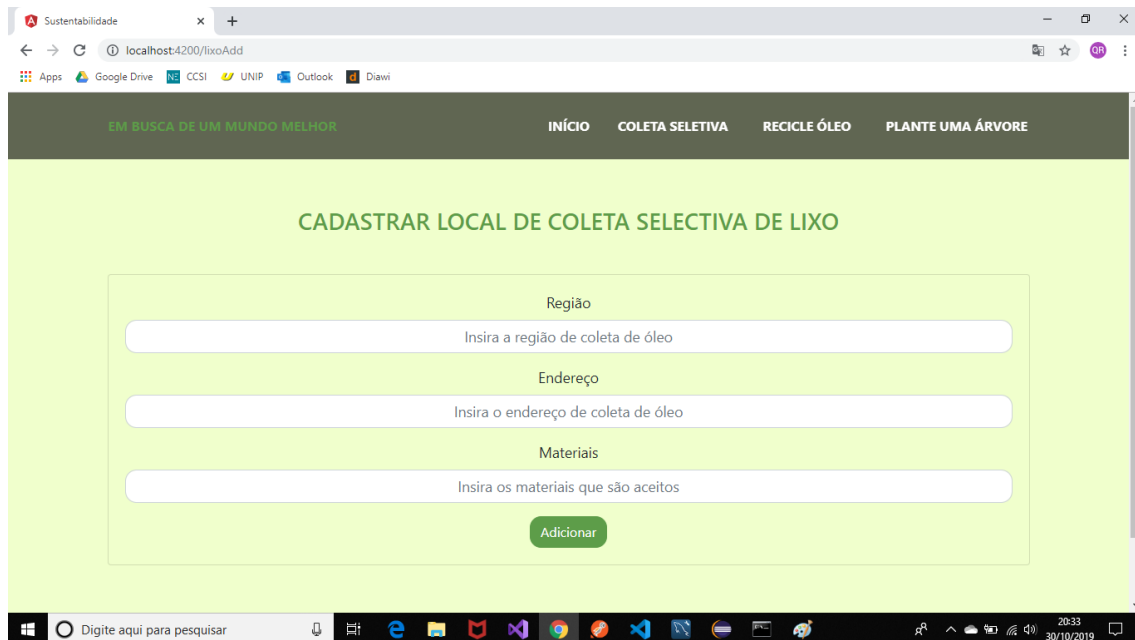
### **5.5. Spring Boot**

O Spring Boot é um projeto da Spring baseado em java criado para facilitar o processo de configuração e publicação de aplicações. Basta setar as configurações desejadas (quais módulos deseja utilizar) para reconhecer e configurar. Um de seus maiores benefícios é que com ele poderia ficar mais livre para pensar nas regras de negócio da aplicação.

## 6. Desenvolvimento

### 6.1. Prints







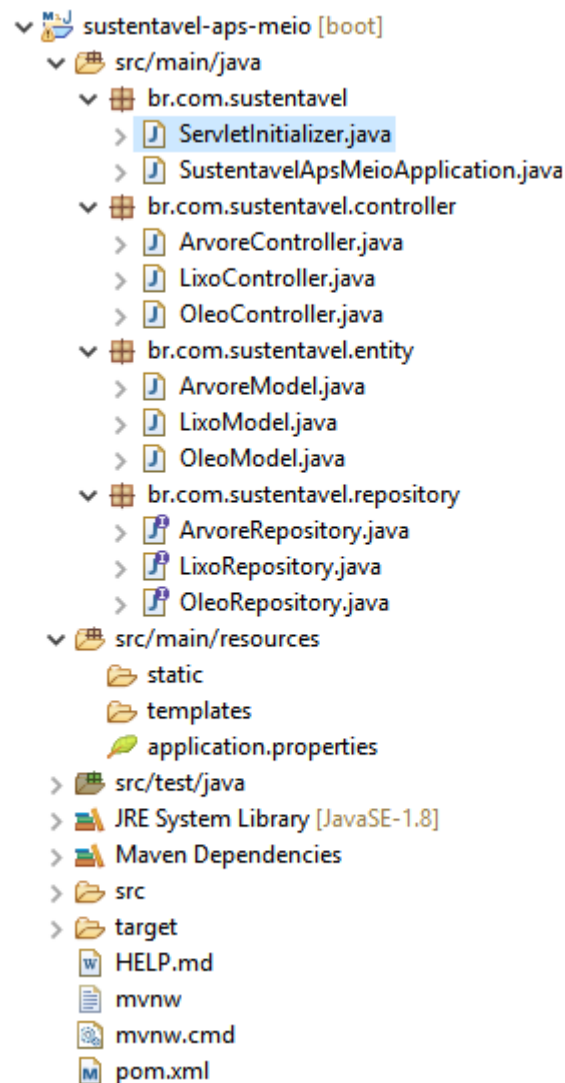


## 6.2. Código

As aplicações se encontram em um repositório github:  
[https://github.com/beevaz/Sustentabilidade\\_APS](https://github.com/beevaz/Sustentabilidade_APS)

### API Spring Boot

Abaixo a organização das classes e interfaces para melhor entendimento.



```
package br.com.sustentavel;
```

```
import org.springframework.boot.builder.SpringApplicationBuilder;
```

```
import
```

```
org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
```

```
public class ServletInitializer extends SpringBootServletInitializer {
```

```
    @Override
```

```

        protected SpringApplicationBuilder configure(SpringApplicationBuilder
application) {
            return application.sources(SustentavelApsMeioApplication.class);
        }
    }
}

```

---

```

package br.com.sustentavel;

```

```

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

```

```

@SpringBootApplication
public class SustentavelApsMeioApplication {

    public static void main(String[] args) {
        SpringApplication.run(SustentavelApsMeioApplication.class,
args);
    }

}

```

---

```

package br.com.sustentavel.controller;

```

```

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.servlet.config.annotation.CorsRegistry;

```

```

import br.com.sustentavel.entity.ArvoreModel;
import br.com.sustentavel.repository.ArvoreRepository;

```

```

@CrossOrigin
@RestController
public class ArvoreController {

    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping("/**")
        .allowedMethods("*")
        .allowedOrigins("http://localhost:4200");
    }
}

```

```

@Autowired
private ArvoreRepository _arvoreRepository;

```

```

    @CrossOrigin
    @RequestMapping(value = "/arvore", method = RequestMethod.GET)
    public List<ArvoreModel> Get() {
        return _arvoreRepository.findAll();
    }
}

```

---

```
package br.com.sustentavel.controller;
```

```
import java.util.List;
import java.util.Optional;
```

```
import javax.validation.Valid;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.servlet.config.annotation.CorsRegistry;
```

```
import br.com.sustentavel.entity.LixoModel;
import br.com.sustentavel.repository.LixoRepository;
```

```

@CrossOrigin
@RestController
public class LixoController {

    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping("/**")
            .allowedMethods("*")
            .allowedOrigins("http://localhost:4200");
    }
}

```

```

@Autowired
private LixoRepository _lixoRepository;

```

```

@CrossOrigin
@RequestMapping(value = "/lixo", method = RequestMethod.GET)
public List<LixoModel> Get() {
    return _lixoRepository.findAll();
}

```

```
@CrossOrigin
```

```

    @RequestMapping(value = "/lixo/{id_lixo}", method = RequestMethod.GET)
    public ResponseEntity<LixoModel> GetById(@PathVariable(value = "id_lixo") long id_lixo)
    {
        Optional<LixoModel> lixo = _lixoRepository.findById(id_lixo);
        if(lixo.isPresent())
            return new ResponseEntity<LixoModel>(lixo.get(), HttpStatus.OK);
        else
            return new ResponseEntity<>(HttpStatus.NOT_FOUND);
    }

    @CrossOrigin
    @RequestMapping(value = "/lixo", method = RequestMethod.POST)
    public LixoModel Post(@Valid @RequestBody LixoModel lixo)
    {
        return _lixoRepository.save(lixo);
    }

    @CrossOrigin
    @RequestMapping(value = "/lixo/{id_lixo}", method = RequestMethod.PUT)
    public ResponseEntity<LixoModel> Put(@PathVariable(value = "id_lixo")
    long id_lixo, @Valid @RequestBody LixoModel newLixo)
    {
        Optional<LixoModel> oldLixo = _lixoRepository.findById(id_lixo);
        if(oldLixo.isPresent()){
            LixoModel lixo = oldLixo.get();
            lixo.setRegiao(newLixo.getRegiao());
            lixo.setEndereco(newLixo.getEndereco());
            _lixoRepository.save(lixo);
            return new ResponseEntity<LixoModel>(lixo, HttpStatus.OK);
        }
        else
            return new ResponseEntity<>(HttpStatus.NOT_FOUND);
    }

    @CrossOrigin
    @RequestMapping(value = "/lixo/{id_lixo}", method =
    RequestMethod.DELETE)
    public ResponseEntity<Object> Delete(@PathVariable(value = "id_lixo") long
    id_lixo)
    {
        Optional<LixoModel> lixo = _lixoRepository.findById(id_lixo);
        if(lixo.isPresent()){
            _lixoRepository.delete(lixo.get());
            return new ResponseEntity<>(HttpStatus.OK);
        }
        else
            return new ResponseEntity<>(HttpStatus.NOT_FOUND);
    }
}

```

---

```

package br.com.sustentavel.controller;

import java.util.List;
import java.util.Optional;

import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.servlet.config.annotation.CorsRegistry;

import br.com.sustentavel.entity.OleoModel;
import br.com.sustentavel.repository.OleoRepository;

@CrossOrigin
@RestController
public class OleoController {

    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping("/**")
            .allowedMethods("*")
            .allowedOrigins("http://localhost:4200");
    }

    @Autowired
    private OleoRepository _oleoRepository;

    @CrossOrigin
    @RequestMapping(value = "/oleo", method = RequestMethod.GET)
    public List<OleoModel> Get() {
        return _oleoRepository.findAll();
    }

    @CrossOrigin
    @RequestMapping(value = "/oleo/{id_oleo}", method = RequestMethod.GET)
    public ResponseEntity<OleoModel> GetById(@PathVariable(value =
"id_oleo") long id_oleo)
    {
        Optional<OleoModel> oleo = _oleoRepository.findById(id_oleo);
        if(oleo.isPresent())
            return new ResponseEntity<OleoModel>(oleo.get(), HttpStatus.OK);
    }

```

```

        else
            return new ResponseEntity<>(HttpStatus.NOT_FOUND);
    }

    @CrossOrigin
    @RequestMapping(value = "/oleo", method = RequestMethod.POST)
    public OleoModel Post(@Valid @RequestBody OleoModel oleo)
    {
        return _oleoRepository.save(oleo);
    }

    @CrossOrigin
    @RequestMapping(value = "/oleo/{id_oleo}", method =
RequestMethod.PUT)
    public ResponseEntity<OleoModel> Put(@PathVariable(value = "id_oleo")
long id_oleo, @Valid @RequestBody OleoModel newOleo)
    {
        Optional<OleoModel> oldOleo = _oleoRepository.findById(id_oleo);
        if(oldOleo.isPresent()){
            OleoModel oleo = oldOleo.get();
            oleo.setRegiao(newOleo.getRegiao());
            oleo.setEndereco(newOleo.getEndereco());
            _oleoRepository.save(oleo);
            return new ResponseEntity<OleoModel>(oleo, HttpStatus.OK);
        }
        else
            return new ResponseEntity<>(HttpStatus.NOT_FOUND);
    }

    @CrossOrigin
    @RequestMapping(value = "/oleo/{id_oleo}", method =
RequestMethod.DELETE)
    public ResponseEntity<Object> Delete(@PathVariable(value = "id_oleo")
long id_oleo)
    {
        Optional<OleoModel> oleo = _oleoRepository.findById(id_oleo);
        if(oleo.isPresent()){
            _oleoRepository.delete(oleo.get());
            return new ResponseEntity<>(HttpStatus.OK);
        }
        else
            return new ResponseEntity<>(HttpStatus.NOT_FOUND);
    }
}

```

---

```

package br.com.sustentavel.entity;

```

```

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;

```

```

import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;

@Entity
@Table(name="arvores")
public class ArvoreModel {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private long id_arvore;
    @NotNull
    private String nome_arvore;
    @NotNull
    private String nome_cientifico;

    //Getters
    public long getId_Arvore() {
        return id_arvore;
    }
    public String getNome_Arvore() {
        return nome_arvore;
    }
    public String getNome_Cientifico() {
        return nome_cientifico;
    }

    public void setId_Arvore(Long id_arvore) {
        this.id_arvore = id_arvore;
    }
    public void setNome_Arvore(String nome_arvore) {
        this.nome_arvore = nome_arvore;
    }
    public void setNome_Cientifico(String nome_cientifico) {
        this.nome_cientifico = nome_cientifico;
    }
}

```

---

```

package br.com.sustentavel.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;

@Entity

```



```

@Table(name="lixo")
public class LixoModel {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private long id_lixo;
    @NotNull
    private String endereco;
    @NotNull
    private String regioao;
    @NotNull
    private String materiais;

    //Getters
    public long getId_Lixo() {
        return id_lixo;
    }
    public String getRegiao() {
        return regioao;
    }
    public String getEndereco() {
        return endereco;
    }
    public String getMateriais() {
        return materiais;
    }

    public void setId_Lixo(Long id_lixo) {
        this.id_lixo = id_lixo;
    }
    public void setRegiao(String regioao) {
        this.regiao = regioao;
    }
    public void setEndereco(String endereco) {
        this.endereco = endereco;
    }
    public void setMateriais(String materiais) {
        this.materiais = materiais;
    }
}

```

---

```

package br.com.sustentavel.entity;

```

```

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;

```

```

@Entity
@Table(name="oleo")

```

```

public class OleoModel {

    @Id
    @NotNull
    private long id_oleo;
    @NotNull
    private String regioao;
    @NotNull
    private String endereco;

    //Getters
    public long getId_Oleo() {
        return id_oleo;
    }
    public String getRegiao() {
        return regioao;
    }
    public String getEndereco() {
        return endereco;
    }

    public void setId_Oleo(Long id_oleo) {
        this.id_oleo = id_oleo;
    }
    public void setRegiao(String regioao) {
        this.regiao = regioao;
    }
    public void setEndereco(String endereco) {
        this.endereco = endereco;
    }
}

```

---

```

package br.com.sustentavel.repository;

```

```

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

```

```

import br.com.sustentavel.entity.ArvoreModel;

```

```

@Repository
public interface ArvoreRepository extends JpaRepository<ArvoreModel, Long> {
}

```

---

```

package br.com.sustentavel.repository;

```

```

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

```

```
import br.com.sustentavel.entity.LixoModel;
```

```
@Repository
```

```
public interface LixoRepository extends JpaRepository<LixoModel, Long> { }
```

---

```
package br.com.sustentavel.repository;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import org.springframework.stereotype.Repository;
```

```
import br.com.sustentavel.entity.OleoModel;
```

```
@Repository
```

```
public interface OleoRepository extends JpaRepository<OleoModel, Long> { }
```

---

```
#INFORMAÇÕES PARA CONEXÃO COM O BANCO DE DADOS
```

```
spring.jpa.hibernate.ddl-auto=update
```

```
spring.datasource.url = jdbc:mysql://dbqrpay.c96j5r4p21xz.us-east-2.rds.amazonaws.com:3306/APS
```

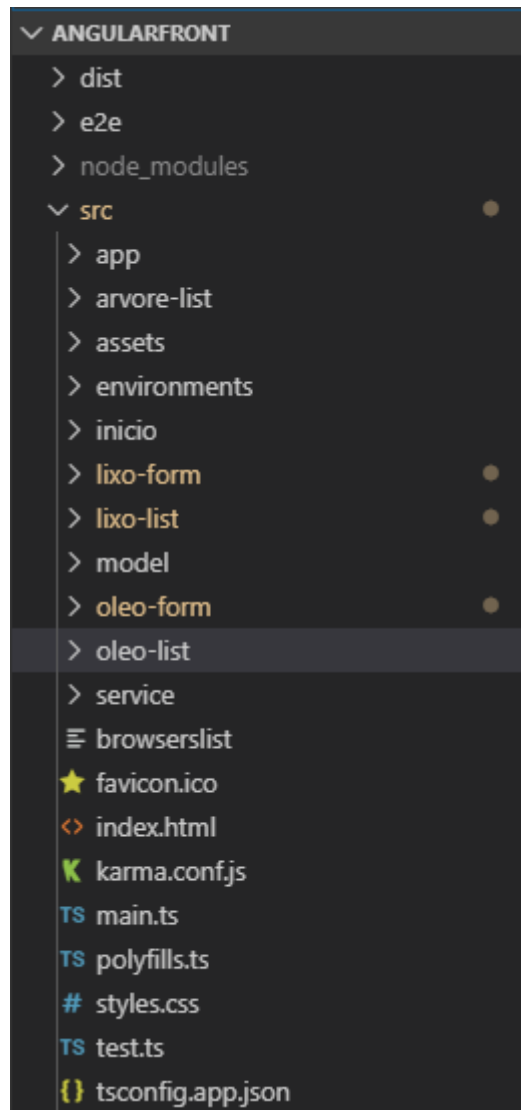
```
spring.datasource.username =root
```

```
spring.datasource.password =12345678
```

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

## Front-End / Consumo Angular JS

Abaixo a organização do web site em Angular JS para melhor entendimento.



```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { ArvoreListComponent } from '../arvore-list/arvore-list.component';
import { OleoListComponent } from '../oleo-list/oleo-list.component';
import { OleoFormComponent } from '../oleo-form/oleo-form.component';
import { LixoListComponent } from '../lixo-list/lixo-list.component';
import { LixoFormComponent } from '../lixo-form/lixo-form.component';
import { InicioComponent } from '../inicio/inicio.component';
```

```
const routes: Routes = [
  { path: 'arvores', component: ArvoreListComponent },
  { path: 'oleos', component: OleoListComponent },
  { path: 'oleoAdd', component: OleoFormComponent },
```

```

    { path: 'lixos', component: LixoListComponent },
    { path: 'lixoAdd', component: LixoFormComponent },
    { path: '', component: InicioComponent }
  ];

```

```

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModuleModule { }

```

---

```

<header>
  <ul>
    <li class="logo"><b>EM BUSCA DE UM MUNDO MELHOR</b></li>
    <li class="list-inline-item"><a routerLink="/arvores" >Plante uma
Árvore</a></li>
    <li class="list-inline-item"><a routerLink="/oleos" >Recicle Óleo</a></li>
    <li class="list-inline-item"><a routerLink="/lixos" >Coleta Seletiva</a></li>
    <li class="list-inline-item"><a routerLink="/" >Início</a></li>
  </ul>
</header>
<div class="container">
  <div class="row">
    <div class="col-md-12">
      <div class="topmargin">
        <router-outlet>

        </router-outlet>
      </div>
    </div>
  </div>
</div>
<footer>
  <p class="Bottom">EM BUSCA DE UM MUNDO MELHOR</p>
  <p class="copy"><b>© Copyright 2019</b></p>
</footer>

```

---

```

import { TestBed, async } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';

```

```

describe('AppComponent', () => {
  beforeEach(async(() => {
    TestBed.configureTestingModule({
      imports: [
        RouterTestingModule
      ],
      declarations: [
        AppComponent
      ],

```

```

    }).compileComponents();
  }));

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.debugElement.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'angularfront'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.debugElement.componentInstance;
    expect(app.title).toEqual('angularfront');
  });

  it('should render title in a h1 tag', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.debugElement.nativeElement;
    expect(compiled.querySelector('h1').textContent).toContain('Welcome to
angularfront!');
  });
});

```

---

```
import { Component } from '@angular/core';
```

```

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {

  title: string;

  constructor() {
    this.title = 'Sustentabilidade';
  }

}

```

---

```

import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

```

```

import { AppRoutingModule } from './app-routing.module';
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { AppComponent } from './app.component';
import { ArvoreListComponent } from '../arvore-list/arvore-list.component';

```

```

import {OleoFormComponent } from '../oleo-form/oleo-form.component';
import { SusService } from '../service/sus-service.service';
import { OleoListComponent } from '../oleo-list/oleo-list.component';
import { LixoListComponent } from '../lixo-list/lixo-list.component';
import { LixoFormComponent } from '../lixo-form/lixo-form.component';
import { InicioComponent } from '../inicio/inicio.component';

```

```

import {BrowserAnimationsModule} from '@angular/platform-
browser/animations';
import { CustomMaterialModule } from './material.module';

```

```

@NgModule({
  declarations: [
    AppComponent,
    ArvoreListComponent,
    OleoFormComponent,
    OleoListComponent,
    LixoFormComponent,
    LixoListComponent,
    InicioComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
    HttpClientModule,
    BrowserAnimationsModule,
    CustomMaterialModule
  ],
  providers: [SusService],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

---

```

import {NgModule} from "@angular/core";
import { CommonModule } from '@angular/common';
import {
  MatButtonModule, MatCardModule, MatDialogModule, MatInputModule,
  MatTableModule,
  MatToolbarModule, MatMenuModule, MatIconModule,
  MatProgressSpinnerModule
} from '@angular/material';
@NgModule({
  imports: [
    CommonModule,
    MatToolbarModule,
    MatButtonModule,
    MatCardModule,
    MatInputModule,

```

```

MatDialogModule,
MatTableModule,
MatMenuModule,
MatIconModule,
MatProgressSpinnerModule
],
exports: [
CommonModule,
MatToolbarModule,
MatButtonModule,
MatCardModule,
MatInputModule,
MatDialogModule,
MatTableModule,
MatMenuModule,
MatIconModule,
MatProgressSpinnerModule
],
})
export class CustomMaterialModule { }

```

---

```

<div class="centralizar">
<h1>{{titulo}}</h1>
<p>
Que tal pensar de uma forma mais sustentável, e ajudar o planeta?<br />
Abaixo listamos as <b>árvores típicas brasileiras</b>, para que você possa
conhecer, e escolher uma para plantá-la.
</p>
<div class="card my-5">
<div class="card-body">
<table class="table table-bordered table-striped">
<thead class="thead-dark">
<tr>
<th scope="col" style="display: none;">ID</th>
<th scope="col">ÁRVORE</th>
<th scope="col">NOME CIENTÍFICO</th>

</tr>
</thead>
<tbody>
<tr *ngFor="let arvore of arvores; let i = index">
<td style="display: none;">{{ arvore.id_Arvore }}</td>
<td>{{ arvore.nome_Arvore }}</td>
<td>{{ arvore.nome_Cientifico }}</td>

</tr>
</tbody>
</table>
</div>

```



</div>  
</div>

---

```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';

import { ArvoreListComponent } from './arvore-list.component';

describe('ArvoreListComponent', () => {
  let component: ArvoreListComponent;
  let fixture: ComponentFixture<ArvoreListComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ ArvoreListComponent ]
    })
    .compileComponents();
  }));

  beforeEach(() => {
    fixture = TestBed.createComponent(ArvoreListComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

---

```
import { Component, OnInit } from '@angular/core';
import { Arvore } from '../model/arvore';
import { SusService } from '../service/sus-service.service';

@Component({
  selector: 'app-arvore-list',
  templateUrl: './arvore-list.component.html',
  styleUrls: ['./arvore-list.component.css']
})
export class ArvoreListComponent implements OnInit {

  private arvores: Arvore[] = new Array();
  private titulo:string;

  constructor(private arvoreService: SusService) {
  }

  ngOnInit() {
    /*Título da Página*/
  }
}
```

```

    this.titulo = "Plante uma Árvore";

    /*Chama o serviço que retorna as árvores cadastradas*/
    this.arvoreService.findAll().subscribe(res => this.arvores = res);
  }
}

```

---

```

<div class="centralizar">
  <div class="imageHome">
    <div class="textoHome">
      <p>
        <b>BEM VINDO AO SUSTENTABILIDADE- EM BUSCA DE UM FUTURO
MELHOR!</b><br/><br/>
        Como você já chegou aqui, temos uma pergunta para você...
      </p>
      <p class="planeta">Vamos ajudar a salvar o nosso planeta?</p>
      <p>
        Esta iniciativa nasceu para que as pessoas, em busca de um futuro
melhor e mais sustentável possam se ajudar<br/>
        Então sinta-se à vontade para navegar entre as páginas e personalizar os
conteúdos.
      <br />
        Lembre-se sempre... esta é uma iniciativa séria, com a finalidade de
auxiliar a todos!
      <br />
        Estamos juntos nessa!
      </p>
    </div>
  </div>
</div>

```

```

<div>
  <h3>Coleta Seletiva</h3>
  <p style="text-align: center;">
    
  </p>
  <p>Na seção descrita como <b>coleta seletiva</b>, trouxemos para você uma
lista de locais que fazem este tipo de trabalho,
  e quais são os materiais que aceitam!
</p>
  <h3>E por que realizar a coleta seletiva?</h3>
  <p>É simples... A Coleta Seletiva é o primeiro e o mais importante passo para
fazer com que vários tipos de
  resíduos sigam seu caminho para reciclagem ou destinação final
ambientalmente correta, pois o resíduo
  separado corretamente deixa de ser lixo, e assim retira a poluição das ruas e
do meio ambiente em geral.</p>
  <hr>
  <h3>Recicle óleo</h3>

```

```

<p style="text-align: center;">

</p>
<p>Você sabia que óleo e água não se misturam? O descarte incorreto deste
pode poluir, rios, mares, lagos e outros.
  A melhor solução é você ir atrás de um correto destino para ele! Que tal
olhar os locais que recebem óleo usado para reciclar?
  Acesse <b>recicle óleo</b> e fique por dentro!
</p>
<hr>
<h3>Plante uma árvore</h3>
<p style="text-align: center;">

</p>
<p>Ao acessar <b>plante uma árvore</b> você poderá visualizar uma grande
parte de árvores típicas brasileiras
  para poder se inspirar e começar essa ação!</p>
  <h3>Por que plantar?</h3>
<p>As árvores ajudam a limpar poluição, "limpando" o ar que respiramos. Ela
transforma o gás carbono em oxigênio!</p>

</div>
</div>

```

---

```

import { async, ComponentFixture, TestBed } from '@angular/core/testing';

import { InicioComponent } from './inicio.component';

describe('InicioComponent', () => {
  let component: InicioComponent;
  let fixture: ComponentFixture<InicioComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ InicioComponent ]
    })
    .compileComponents();
  }));

  beforeEach(() => {
    fixture = TestBed.createComponent(InicioComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

---

```

import { Component, OnInit } from '@angular/core';
import { Lixo } from '../model/lixo';
import { Router } from '@angular/router';
import { SusService } from '../service/sus-service.service';

@Component({
  selector: 'app-inicio',
  templateUrl: './inicio.component.html',
  styleUrls: ['./inicio.component.css']
})
export class InicioComponent implements OnInit {

  private lixos: Lixo[] = new Array();
  private titulo:string;

  constructor(private lixoService: SusService, private router: Router) {
  }

  ngOnInit() {
  }
}

```

---

```

<h3>Cadastrar local de coleta selectiva de lixo</h3>
<div class="card my-5">
  <div class="card-body">
    <form name="form1" (ngSubmit)="onSubmit()" #lixoForm="ngForm"
method="POST">

      <div class="form-group">
        <label for="regiao">Região</label>
        <input type="text" [(ngModel)]="lixo.regiao" class="form-control"
id="regiao" name="regiao"
placeholder="Insira a região de coleta de óleo" required
#name="ngModel" />
        <div [hidden]="!regiao.pristine"></div>
      </div>

      <div class="form-group">
        <label for="endereco">Endereço</label>
        <input type="text" [(ngModel)]="lixo.endereco" class="form-control"
id="endereco" name="endereco"
placeholder="Insira o endereço de coleta de óleo" required
#endereco="ngModel" >
      </div>

```

```

    <div class="form-group">
      <label for="materiais">Materiais</label>
      <input type="text"[(ngModel)]="lixo.materiais" class="form-control"
id="materiais" name="materiais"
      placeholder="Insira os materiais que são aceitos" required
#endereco="ngModel" >
    </div>

    <button type="submit" [disabled]="!lixoForm.form.valid" class="btn btn-
info" id="bt"> Adicionar</button>
  </form>
</div>
</div>

```

---

```

import { async, ComponentFixture, TestBed } from '@angular/core/testing';

import { LixoFormComponent } from './lixo-form.component';

describe('LixoFormComponent', () => {
  let component: LixoFormComponent;
  let fixture: ComponentFixture<LixoFormComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ LixoFormComponent ]
    })
    .compileComponents();
  }));

  beforeEach(() => {
    fixture = TestBed.createComponent(LixoFormComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

---

```

import { Component, Directive, HostListener, Input } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { SusService } from '../service/sus-service.service';
import { Lixo } from '../model/lixo';
import { ControlValueAccessor } from '@angular/forms';

@Component({
  selector: 'app-lixo-form',

```

```

    templateUrl: './lixo-form.component.html',
    styleUrls: ['./lixo-form.component.css']
  })

  export class LixoFormComponent {

    lixo: Lixo;

    constructor(private route: ActivatedRoute, private router: Router, private
    lixoService: SusService) {
      this.lixo = new Lixo();
    }

    /*Quando clicar em adicionar*/
    onSubmit() {
      var regioao = document.getElementById("regiao") as HTMLInputElement;
      var endereco = document.getElementById("endereco") as
      HTMLInputElement;

      if (regiao.value == "") {
        alert("Preencha o campo com a região");
      } else if (endereco.value == "") {
        alert("Preencha o campo com o endereço");
      } else {
        this.lixoService.saveLixo(this.lixo).subscribe(result => this.gotoUserList());

        alert("Local de coleta de lixo cadastrado com sucesso");

      }
    }
    /*Retorna para página da listagem de locais de coleta de lixo*/
    gotoUserList() {
      this.router.navigate(['/lixos']);
    }

  }

```

---

```

<div class="centralizar">
<h1>{{titulo}}</h1>
<p>
  Trouxemos para você uma lista de locais que fazem coleta de lixo seletiva
  para reciclagem!<br />
  Nosso intuito é que sejamos um canal interativo, sendo assim, você como
  usuário, pode:<br><br>
  - <b>Incluir</b>, algum local que ainda não esteja aqui cadastrado.<br>
  - <b>Alterar</b>, se algum destes locais tiveram alguma modificação.<br>
  - <b>Excluir</b>, caso não esteja mais fazendo este tipo de atendimento.<br>
</p>

```

```

<p><a (click)="novo();" class="referencia_link" style="cursor: pointer;">CLIQUE
AQUI para cadastrar um novo local de coleta!</a></p>
<div class="card my-5">
  <div class="card-body">
    <table class="table table-bordered table-striped">
      <thead class="thead-dark">
        <tr>
          <th scope="col" style="display: none;">ID</th>
          <th scope="col" class="tamanhoInput">REGIÃO</th>
          <th scope="col">ENDEREÇO</th>
          <th scope="col">MATERIAIS</th>
          <th scope="col" class="tamanhoInput">EXCLUIR</th>
          <th scope="col" class="tamanhoInput">SALVAR</th>

        </tr>
      </thead>
      <tbody>
        <tr *ngFor="let lixo of lixos; let i = index">
          <td style="display: none;"><input [(ngModel)]= "lixo.id_Lixo" disabled
id="Id_Oleo"/></td>
          <td class="tamanhoInput"><textarea [(ngModel)]= "lixo.regiao"
id="RegiaoOleo" rows="2"></textarea></td>
          <td><textarea [(ngModel)]= "lixo.endereco" id="EnderecoOleo"
rows="2"></textarea></td>
          <td><textarea [(ngModel)]= "lixo.materiais" id="EnderecoOleo"
rows="2"></textarea></td>

          <td class="tamanhoInput"><a (click)="excluir(lixo);" style="cursor:
pointer;">Excluir</a></td>
          <td class="tamanhoInput"><a (click)="salvar(lixo);" style="cursor: pointer;"
>Salvar Alteração</a></td>
        </tr>
      </tbody>
    </table>
  </div>

```

---

```

import { async, ComponentFixture, TestBed } from '@angular/core/testing';

import { LixoListComponent } from './lixo-list.component';

describe('LixoListComponent', () => {
  let component: LixoListComponent;
  let fixture: ComponentFixture<LixoListComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ LixoListComponent ]
    })
    .compileComponents();

```

```

    ));

    beforeEach(() => {
        fixture = TestBed.createComponent(LixoListComponent);
        component = fixture.componentInstance;
        fixture.detectChanges();
    });

    it('should create', () => {
        expect(component).toBeTruthy();
    });
});

```

---

```

import { Component, OnInit } from '@angular/core';
import { Lixo } from '../model/lixo';
import { Router } from '@angular/router';
import { SusService } from '../service/sus-service.service';

@Component({
    selector: 'app-lixo-list',
    templateUrl: './lixo-list.component.html',
    styleUrls: ['./lixo-list.component.css']
})
export class LixoListComponent implements OnInit {

    private lixos: Lixo[] = new Array();
    private titulo:string;

    constructor(private lixoService: SusService, private router: Router) {
    }

    ngOnInit() {

        /*Título da Página*/
        this.titulo = "Recicle o que você não vai mais usar!";
        /*Chama o serviço que retorna todos os locais de coleta de lixo
        cadastrados*/
        this.lixoService.findAllLixo().subscribe(res => this.lixos = res);
    }

    /*Exclui um registro quando clicar em excluir*/
    excluir(lixo:Lixo):void {

        var id = lixo.id_Lixo;
        if(confirm("Deseja realmente excluir esse registro?")){

            /*Chama serviço que exclui o local de coleta de lixo*/
            this.lixoService.deleteLixo(id).subscribe(response => {

```



```

        /*Pega a resposta do serviço*/
        let res:Response = <Response>response;
        alert("Item Excluído com sucesso");
        window.location.reload();
        this.router.navigate(['/lixos']);
    },
    (erro) => {
        /*Caso tenha algum erro*/
        alert(erro);
    });
}

}

novo():void{

    this.router.navigate(['/lixoAdd']);

}

/*Mesmo padrão que o excluir, mas salvando as alterações de edição*/
salvar(lixo:Lixo):void{
    if(confirm("Deseja realmente alterar esse registro?")){
        this.lixoService.updateLixo(lixo).subscribe(response => {

            let res:Response = <Response>response;
            alert("Item Alterado com sucesso");
            window.location.reload();
            this.router.navigate(['/lixos']);
        },
        (erro) => {

            alert(erro);
        });
    }
}
}
}

```

---

```

import { Arvore } from './arvore';

describe('Arvore', () => {
    it('should create an instance', () => {
        expect(new Arvore()).toBeTruthy();
    });
});

```

---

```

export class Arvore {

```

```
id_Arvore: number;
nome_Arvore: string;
nome_Cientifico: string;
}
```

---

```
import { Lixo } from './lixo';
```

```
describe('Lixo', () => {
  it('should create an instance', () => {
    expect(new Lixo()).toBeTruthy();
  });
});
```

---

```
export class Lixo {
  id_Lixo: number;
  regioao: string;
  endereco: string;
  materiais: string;
}
```

---

```
import { Oleo } from './oleo';
```

```
describe('Oleo', () => {
  it('should create an instance', () => {
    expect(new Oleo()).toBeTruthy();
  });
});
```

---

```
export class Oleo {
  id_Oleo: number;
  regioao: string;
  endereco: string;
}
```

---

```
<h3>Cadastrar local de coleta de óleo</h3>
```

```
<div class="card my-5">
```

```
  <div class="card-body">
```

```
    <form name="form1" (ngSubmit)="onSubmit()" #oleoForm="ngForm"
method="POST">
```

```
      <div class="form-group">
```

```
        <label for="regiao"><b>REGIÃO</b></label>
```

```
        <input type="text" [(ngModel)]="oleo.regiao" class="form-control"
id="regiao" name="regiao"
```

```
          placeholder="Insira a região de coleta de óleo" required
#name="ngModel" />
```

```

    <div [hidden]="!regiao.pristine"></div>
  </div>

  <div class="form-group">
    <label for="endereco"><b>ENDEREÇO</b></label>
    <input type="text"[(ngModel)]="oleo.endereco" class="form-control"
id="endereco" name="endereco"
    placeholder="Insira o endereço de coleta de óleo" required
#endereco="ngModel" >
  </div>

  <button type="submit" [disabled]="!oleoForm.form.valid" class="btn btn-
info" id="bt"> Adicionar</button>
</form>
</div>
</div>

```

---

```

import { async, ComponentFixture, TestBed } from '@angular/core/testing';

import { OleoFormComponent } from './oleo-form.component';

describe('OleoFormComponent', () => {
  let component: OleoFormComponent;
  let fixture: ComponentFixture<OleoFormComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ OleoFormComponent ]
    })
    .compileComponents();
  }));

  beforeEach(() => {
    fixture = TestBed.createComponent(OleoFormComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

---

```

import { Component, Directive, HostListener, Input } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { SusService } from '../service/sus-service.service';
import { Oleo } from '../model/oleo';

```

```

import { ControlValueAccessor } from '@angular/forms';

@Component({
  selector: 'app-oleo-form',
  templateUrl: './oleo-form.component.html',
  styleUrls: ['./oleo-form.component.css']
})

export class OleoFormComponent {

  oleo: Oleo;

  constructor(private route: ActivatedRoute, private router: Router, private
oleoService: SusService) {
    this.oleo = new Oleo();
  }

  onSubmit() {
    var regiao = document.getElementById("regiao") as HTMLInputElement;
    var endereco = document.getElementById("endereco") as
HTMLInputElement;

    if (regiao.value == "") {
      alert("Preencha o campo com a região");
    } else if (endereco.value == "") {
      alert("Preencha o campo com o endereço");
    } else {
      this.oleoService.saveOleo(this.oleo).subscribe(result => this.gotoUserList());

      alert("Local de coleta de óleo cadastrado com sucesso");
    }
  }
  gotoUserList() {
    this.router.navigate(['/oleos']);
  }
}

```

---

```

<div class="centralizar">
<h1>{{titulo}}</h1>
<p>
  Trouxemos para você uma lista de locais para reciclagem de óleo!<br />
  Nosso intuito é que sejamos um canal interativo, sendo assim, você como
  usuário, pode:<br><br>
  - <b>Incluir</b>, algum local que ainda não esteja aqui cadastrado.<br>
  - <b>Alterar</b>, se algum destes locais tiveram alguma modificação.<br>
  - <b>Excluir</b>, caso não esteja mais fazendo este tipo de atendimento.<br>
</p>

```

<p><a (click)="novo();" class="referencia\_link" style="cursor: pointer;">CLIQUE AQUI para cadastrar um novo local de coleta!</a></p>

<div class="card my-5">

<div class="card-body">

<table class="table table-bordered table-striped">

<thead class="thead-dark">

<tr>

<th scope="col" style="display: none;">ID</th>

<th scope="col" class="tamanhoInput">REGIÃO</th>

<th scope="col">ENDEREÇO</th>

<th scope="col" class="tamanhoInput">EXCLUIR</th>

<th scope="col" class="tamanhoInput">SALVAR</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let oleo of oleos; let i = index">

<td style="display: none;"><input [(ngModel)]="oleo.id\_Oleo" disabled id="Id\_Oleo"/></td>

<td class="tamanhoInput"><input [(ngModel)]="oleo.regiao" id="RegiaoOleo" /></td>

<td><input [(ngModel)]="oleo.endereco" id="EnderecoOleo"/></td>

<td class="tamanhoInput"><a (click)="excluir(oleo);" style="cursor: pointer;">Excluir</a></td>

<td class="tamanhoInput"><a (click)="salvar(oleo);" style="cursor: pointer;">Salvar Alteração</a></td>

</tr>

</tbody>

</table>

</div>

---

```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';
```

```
import { OleoListComponent } from './oleo-list.component';
```

```
describe('OleoListComponent', () => {  
  let component: OleoListComponent;  
  let fixture: ComponentFixture<OleoListComponent>;
```

```
  beforeEach(async() => {  
    TestBed.configureTestingModule({  
      declarations: [ OleoListComponent ]  
    })  
    .compileComponents();  
  }));
```

```
  beforeEach(() => {  
    fixture = TestBed.createComponent(OleoListComponent);
```

```

    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

---

```

import { Component, OnInit } from '@angular/core';
import { Oleo } from '../model/oleo';
import { Router } from '@angular/router';
import { SusService } from '../service/sus-service.service';

@Component({
  selector: 'app-oleo-list',
  templateUrl: './oleo-list.component.html',
  styleUrls: ['./oleo-list.component.css']
})
export class OleoListComponent implements OnInit {

  private oleos: Oleo[] = new Array();
  private titulo:string;

  constructor(private oleoService: SusService, private router: Router) {
  }

  /*Mesmo padrão da página de coleta de lixo, caso dúvidas, olhar os
  comentários lá*/
  ngOnInit() {
    this.titulo = "Recicle Óleo";

    this.oleoService.findAllOleo().subscribe(res => this.oleos = res);
  }

  excluir(oleo:Oleo):void {

    var id = oleo.id_Oleo;
    if(confirm("Deseja realmente excluir esse registro?")){

      this.oleoService.deleteOleo(id).subscribe(response => {

        let res:Response = <Response>response;
        alert("Item Excluído com sucesso");
        window.location.reload();
        this.router.navigate(['/oleos']);
      },
      (erro) => {
        alert(erro);
      }
    )
  }
}

```

```

    });
  }

}

novo():void{

  this.router.navigate(['/oleoAdd']);

}

salvar(oleo:Oleo):void{
  if(confirm("Deseja realmente alterar esse registro?")){
    this.oleoService.updateOleo(oleo).subscribe(response => {

      let res:Response = <Response>response;
      alert("Item Alterado com sucesso");
      window.location.reload();
      this.router.navigate(['/oleos']);
    },
    (erro) => {
      alert(erro);
    });
  }
}
}

```

---

```

import { TestBed } from '@angular/core/testing';

import { SusService } from './sus-service.service';

describe('SusServiceService', () => {
  beforeEach(() => TestBed.configureTestingModule({}));

  it('should be created', () => {
    const service: SusService = TestBed.get(SusService);
    expect(service).toBeTruthy();
  });
});

```

---

```

import { Injectable } from '@angular/core';
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { Arvore } from '../model/arvore';
import { Oleo } from '../model/oleo';
import { Lixo } from '../model/lixo';
import { Observable } from 'rxjs/Observable';

@Injectable()

```

```

export class SusService {

    private arvoresUrl: string;
    private oleosUrl: string;
    private lixosUrl: string;

    constructor(private http: HttpClient) {
        this.arvoresUrl = 'http://localhost:8080/arvore';
        this.oleosUrl = 'http://localhost:8080/oleo';
        this.lixosUrl = 'http://localhost:8080/lixo';
    }

    /*Serviço para buscar todas as árvores*/
    public findAll(): Observable<Arvore[]> {
        return this.http.get<Arvore[]>(this.arvoresUrl);
    }

    /*Serviço para buscar todos locais de coleta de óleo*/
    public findAllOleo(): Observable<Oleo[]> {
        return this.http.get<Oleo[]>(this.oleosUrl);
    }

    /*Serviço para buscar um local de coleta de óleo específico*/
    public getOleo(id_Oleo:number): Observable<Oleo[]> {
        return this.http.get<Oleo[]>(`${this.oleosUrl}/${id_Oleo}`);
    }

    /*Serviço para criar um local de coleta de óleo*/
    public saveOleo(oleo: Oleo) {
        return this.http.post<Oleo>(this.oleosUrl , oleo);
    }

    /*Serviço para deletar um local de coleta de óleo*/
    public deleteOleo(id_Oleo: number) : Observable<any>{
        return this.http.delete(`${this.oleosUrl}/${id_Oleo}`);
    }

    /*Serviço para atualizar local de coleta de óleo*/
    public updateOleo(oleo:Oleo) : Observable<any>{
        return this.http.put<Oleo>(`${this.oleosUrl}/${oleo.id_Oleo}`, oleo);
    }

    /*Serviço para buscar todas os locais de coleta de lixo*/
    public findAllLixo(): Observable<Lixo[]> {
        return this.http.get<Lixo[]>(this.lixosUrl);
    }

    /*Serviço para buscar um local de coleta de lixo específico*/
    public getLixo(id_Lixo:number): Observable<Lixo[]> {

```



```

    return this.http.get<Lixo[]>(` ${this.lixosUrl}/${id_Lixo}`);
  }

  /*Serviço para criar um local de coleta de lixo*/
  public saveLixo(lixo: Lixo) {
    return this.http.post<Lixo>(this.lixosUrl , lixo);
  }

  /*Serviço para deletar um local de coleta de lixo*/
  public deleteLixo(id_Lixo: number) : Observable<any>{
    return this.http.delete(` ${this.lixosUrl}/${id_Lixo}`);
  }

  /*Serviço para alterar um local de coleta de lixo*/
  public updateLixo(lixo:Lixo) : Observable<any>{
    return this.http.put<Lixo>(` ${this.lixosUrl}/${lixo.id_Lixo}`, lixo);
  }
}

```

---

```

/* You can add global styles to this file, and also import other style files */

```

```

@import '~@angular/material/prebuilt-themes/indigo-pink.css';

```

```

body {
  margin: 0;
}

```

```

header {
  position: fixed;
  right: 0;
  left: 0;
  padding-right: 6vw;
  padding-left: 1.8vw;
  z-index: 1030;
  background-color: rgba(0,0,0,0.6);
  transition: background-color .2s linear;
  min-height: 80px;
}

```

```

header ul {
  list-style-type: none;
  margin: 0px 15px 0px 0px;
  padding: 0;
  overflow: hidden;
  color: #FFFFFF;
  min-height: 80px;
}

```

```

header li {

```

```

display: block;
text-align: center;
text-decoration: none;
color: #FFFFFF;
padding: 28px 17px;
float: right;
}

header li a {
display: block;
text-align: center;
text-decoration: none;
color: white;
text-transform: uppercase;
background-color: transparent;
font-weight: bold;
}

header li a:hover:not(.active) {
color: #5d9d49;
display: block;
text-align: center;
text-decoration: none;
text-transform: uppercase;
background-color: transparent;
font-weight: bold;
}

.topmargin {
height: 10vw;
}

td, th {
text-align: center;
vertical-align: middle!important;
}

input {
width: -webkit-fill-available;
text-align: center;
vertical-align: middle;
border-radius: 1vw;
}

textarea {
width: -webkit-fill-available;
text-align: center;
vertical-align: middle;
}

```

```

.tamanhoInput {
  width: 11vw!important;
}

h1 {
  text-transform: uppercase;
  font-weight: 4vw;
  color: #5d9d49;
  text-align: center;
}

h3 {
  text-transform: uppercase;
  font-weight: 3vw;
  color: #5d9d49;
  text-align: center;
}

.logo {
  float: left;
  padding-left: 7vw;
  color: #5d9d49;
}

body {
  background-color: #F0FFCC;
  padding-bottom: 2vw;
}

.centralizar p {
  text-align: center;
  font-size: 1.3vw;
}

.planeta {
  font-size: 2vw;
  font-weight: bold;
  color: #5d9d49;
}

img {
  width: 25vw;
  margin: 2vw;
}

.referencia_link {
  text-decoration: underline!important;
}

.card {

```

```

    word-wrap: break-word!important;
    background-color: transparent!important;
    background-clip: none!important;
}

form {
    text-align: center;
    font-size: 1.3vw;
}

form b{
    color: #5d9d49;
}

footer {
    background-color: #000000!important;
    min-height: 4vw!important;
    /* bottom: 0!important; */
    width: 100%!important;
    padding-top: 1.2vw!important;
    margin-top: 3vw!important;
    position: absolute!important;
    vertical-align: bottom;
}

.copy {
    text-align: right;
    color: #FFFFFFF;
    font-size: 1vw;
    padding-right: 7vw;
}

.Bottom {
    float: left;
    padding-left: 7vw;
    color: #FFFFFFF!important;
    font-weight: bold;
}

.imageHome {
    background-image: url("/assets/imagens/imageHome.png");
    min-height: 40vw;
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: center;
    margin-top: -10vw;
    margin-left: -8.7vw;
    margin-right: -8.7vw;
    margin-bottom: 3vw;
}

```

```

.imageHome p {
  text-align: center;
  margin: 0;
  padding: 2vw;
  color: #FFFFFF;
  background-color: rgba(0,0,0,0.3);
}

.textoHome {
  padding: 10vw 13vw 7vw 13vw;
}

.table-bordered td, .table-bordered th {
  border: 1px solid #cccccc!important;
}

td a {
  font-weight: bold;
}

td a:hover {
  text-decoration: none!important;
  font-weight: bold;
  color: #5d9d49!important;
}

.form-control {
  display: block;
  width: 100%;
  padding: .375rem .75rem;
  font-size: 1.3vw!important;
  line-height: 1.5;
  color: #495057;
  background-color: #fff;
  background-clip: padding-box;
  border: 1px solid #ced4da;
  border-radius: 1vw!important;
  transition: border-color .15s ease-in-out,box-shadow .15s ease-in-out;
}

.btn-info{
  color: #fff;
  background-color: #5d9d49!important;
  border-color: #5d9d49!important;
  border-radius: 1vw!important;
}

```

## 7. Conclusão

O trabalho possibilitou que entendêssemos a importância de iniciativas sustentáveis, vinculando isso ao desenvolvimento de um sistema distribuído, o que demonstrou que ao “quebrar” o sistema em partes, deixando ele de forma distribuída, para quem quiser ter acesso e desenvolver sobre o conteúdo, era só utilizar o mesmo padrão de comunicação e mensagens enviadas.

Para a parte escrita, visamos destacar as motivações da escolha do tema, e também conceituar as linguagens de programação utilizadas para o desenvolvimento das APIs e web site. Trazendo imagens do resultado final da aplicação, e também o código fonte que resultou neste.

O desenvolvimento do sistema nos auxiliou a entender, e conceituar na prática como funciona todo conteúdo visto em sala ao longo do semestre na disciplina de desenvolvimento de sistemas distribuídos.

Como mensagem final, aliado com nosso tema, queremos deixar a seguinte frase:

*“Semear ideias ecológicas e plantar sustentabilidade é ter a garantia de colhermos um futuro fértil e consciente.”*

*Por. Sivaldo Filho*

## 8. Ficha de Horas

**UNIP**  
UNIVERSIDADE PAULISTA

**FICHA DAS ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS**

NOME: Beatriz Fialho Vaz TURMA: SI7P08 RA: N925EF1  
CURSO: Sistemas de Informações CAMPUS: Chácara II SEMESTRE: 8º Semestre TURNO: Noturno  
CÓDIGO DA ATIVIDADE: Desenvolvimento de Sistemas Distribuídos SEMESTRE: 8º ANO GRADE: 2019

DATA DA ATIVIDADE	DESCRIÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
14/09/2019	Definição do tema da APS	6			
15/09/2019	Pesquisa sobre o tema	6			
21/09/2019	Pesquisa sobre API Spring Boot	6			
22/09/2019	Pesquisa sobre Angular JS	6			
23/09/2019	Desenvolvimento API	8			
28/09/2019	Desenvolvimento API	7			
05/10/2019	Desenvolvimento API	8			
06/10/2019	Desenvolvimento Front-Angular	6			
12/10/2019	Desenvolvimento Front-Angular	8			
19/10/2019	Desenvolvimento Front-Angular	5			
26/10/2019	Testes aplicação	6			
29/10/2019	Testes aplicação	4			
30/10/2019	Revisão de Layout	3			
31/10/2019	Ajustes Documentação	2			
01/11/2019	Preenchimento Ficha APS	1			

(1) Horas atribuídas de acordo com o regulamento das Atividades Práticas Supervisionadas do curso.

TOTAL DE HORAS ATRIBUÍDAS: 82 horas

AValiação: \_\_\_\_\_

Aprovado ou Reprovado

NOTA: \_\_\_\_\_

DATA: 01/11/2019

CARIMBO E ASSINATURA DO COORDENADOR DO CURSO

**UNIP**  
UNIVERSIDADE PAULISTA

**FICHA DAS ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS**

NOME: Giovanna Rebeca Gerlach TURMA: SI7P08 RA: C9494I2  
CURSO: Sistemas de Informações CAMPUS: Chácara II SEMESTRE: 8º Semestre TURNO: Noturno  
CÓDIGO DA ATIVIDADE: Desenvolvimento de Sistemas Distribuídos SEMESTRE: 8º ANO GRADE: 2019

DATA DA ATIVIDADE	DESCRIÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
14/09/2019	Definição do tema da APS	6			
15/09/2019	Pesquisa sobre o tema	6			
21/09/2019	Pesquisa sobre API Spring Boot	6			
22/09/2019	Pesquisa sobre Angular JS	6			
23/09/2019	Desenvolvimento API	8			
28/09/2019	Desenvolvimento API	7			
05/10/2019	Desenvolvimento API	8			
06/10/2019	Desenvolvimento Front-Angular	6			
12/10/2019	Desenvolvimento Front-Angular	8			
19/10/2019	Desenvolvimento Front-Angular	5			
26/10/2019	Testes aplicação	6			
29/10/2019	Testes aplicação	4			
30/10/2019	Revisão de Layout	3			
31/10/2019	Ajustes Documentação	2			
01/11/2019	Preenchimento Ficha APS	1			

(1) Horas atribuídas de acordo com o regulamento das Atividades Práticas Supervisionadas do curso.

TOTAL DE HORAS ATRIBUÍDAS: 82 horas

AValiação: \_\_\_\_\_

Aprovado ou Reprovado

NOTA: \_\_\_\_\_

DATA: 1/11/2019

CARIMBO E ASSINATURA DO COORDENADOR DO CURSO

FICHA DAS ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

NOME: Adriana Ferreira Moreira TURMA: SI7P08 RA: D060CA2  
CURSO: Sistemas de Informações CAMPUS: Chácara II SEMESTRE: 8º Semestre TURNO: Noturno  
CÓDIGO DA ATIVIDADE: Desenvolvimento de Sistemas Distribuídos SEMESTRE: 8º ANO GRADE: 2019

DATA DA ATIVIDADE	DESCRIÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
14/09/2019	Definição do tema da APS	6	<i>[assinatura]</i>		
15/09/2019	Pesquisa sobre o tema	6	<i>[assinatura]</i>		
21/09/2019	Pesquisa sobre API Spring Boot	6	<i>[assinatura]</i>		
22/09/2019	Pesquisa sobre Angular JS	6	<i>[assinatura]</i>		
23/09/2019	Desenvolvimento API	8	<i>[assinatura]</i>		
28/09/2019	Desenvolvimento API	7	<i>[assinatura]</i>		
05/10/2019	Desenvolvimento API	8	<i>[assinatura]</i>		
06/10/2019	Desenvolvimento Front-Angular	6	<i>[assinatura]</i>		
12/10/2019	Desenvolvimento Front-Angular	8	<i>[assinatura]</i>		
19/10/2019	Desenvolvimento Front-Angular	5	<i>[assinatura]</i>		
26/10/2019	Testes aplicação	6	<i>[assinatura]</i>		
29/10/2019	Testes aplicação	4	<i>[assinatura]</i>		
30/10/2019	Revisão de Layout	3	<i>[assinatura]</i>		
31/10/2019	Ajustes Documentação	2	<i>[assinatura]</i>		
01/11/2019	Preenchimento Ficha APS	1	<i>[assinatura]</i>		

(1) Horas atribuídas de acordo com o regulamento das Atividades Práticas Supervisionadas do curso.

TOTAL DE HORAS ATRIBUÍDAS: 82 horas

AValiação: \_\_\_\_\_

Aprovado ou Reprovado

NOTA: \_\_\_\_\_

DATA: 01 / 11 / 2019

CARIMBO E ASSINATURA DO COORDENADOR DO CURSO

FICHA DAS ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

NOME: Laís Dias C. de Araújo TURMA: SI7P08 RA: C976571  
CURSO: Sistemas de Informações CAMPUS: Chácara II SEMESTRE: 8º Semestre TURNO: Noturno  
CÓDIGO DA ATIVIDADE: Desenvolvimento de Sistemas Distribuídos SEMESTRE: 8º ANO GRADE: 2019

DATA DA ATIVIDADE	DESCRIÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
14/09/2019	Definição do tema da APS	6	<i>[assinatura]</i>		
15/09/2019	Pesquisa sobre o tema	6	<i>[assinatura]</i>		
21/09/2019	Pesquisa sobre API Spring Boot	6	<i>[assinatura]</i>		
22/09/2019	Pesquisa sobre Angular JS	6	<i>[assinatura]</i>		
23/09/2019	Desenvolvimento API	8	<i>[assinatura]</i>		
28/09/2019	Desenvolvimento API	7	<i>[assinatura]</i>		
05/10/2019	Desenvolvimento API	8	<i>[assinatura]</i>		
06/10/2019	Desenvolvimento Front-Angular	6	<i>[assinatura]</i>		
12/10/2019	Desenvolvimento Front-Angular	8	<i>[assinatura]</i>		
19/10/2019	Desenvolvimento Front-Angular	5	<i>[assinatura]</i>		
26/10/2019	Testes aplicação	6	<i>[assinatura]</i>		
29/10/2019	Testes aplicação	4	<i>[assinatura]</i>		
30/10/2019	Revisão de Layout	3	<i>[assinatura]</i>		
31/10/2019	Ajustes Documentação	2	<i>[assinatura]</i>		
01/11/2019	Preenchimento Ficha APS	1	<i>[assinatura]</i>		

(1) Horas atribuídas de acordo com o regulamento das Atividades Práticas Supervisionadas do curso.

TOTAL DE HORAS ATRIBUÍDAS: 82 horas

AValiação: \_\_\_\_\_

Aprovado ou Reprovado

NOTA: \_\_\_\_\_

DATA: 01 / 11 / 2019

CARIMBO E ASSINATURA DO COORDENADOR DO CURSO



## 9. Bibliografia

*Developer Mozilla. O que é JavaScript?. Disponível em:*  
<[https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First\\_steps/O\\_que\\_e\\_JavaScript](https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/O_que_e_JavaScript)>.  
*Acesso em: 12 setembro. 2019.*

*Giancarlo Silva. O que é e como funciona a linguagem JavaScript?. Disponível em:* <<https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>>.  
*Acesso em: 12 setembro. 2019.*

*DEV MEDIA. Implementando serviços com AngularJS. Disponível em:* <<https://www.devmedia.com.br/implementando-servicos-com-angularjs/32715/>>.  
*Acesso em: 12 setembro. 2019.*

*Wikipedia. Web API. Disponível em:* <[https://pt.wikipedia.org/wiki/Web\\_API/](https://pt.wikipedia.org/wiki/Web_API/)>.  
*Acesso em: 12 setembro. 2019.*

*Java. O que é a Tecnologia Java e porque preciso dela?. Disponível em:* <[https://www.java.com/pt\\_BR/download/faq/whatis\\_java.xml/](https://www.java.com/pt_BR/download/faq/whatis_java.xml/)>.  
*Acesso em: 12 setembro. 2019.*

*Alexandre Afonso. O que é Spring Boot?. Disponível em:* <<https://blog.algaworks.com/spring-boot/>>.  
*Acesso em: 12 setembro. 2019.*

*Mayra Rosa. Saiba onde estão os pontos de coleta de óleo de cozinha em S. Disponível em:* <<https://ciclovivo.com.br/planeta/meio-ambiente/saiba-onde-estao-os-pontos-de-coleta-de-oleo-de-cozinha-em-sp/>>.  
*Acesso em: 26 outubro. 2019.*

*Do Portal do Governo. Aprenda a descartar o óleo de cozinha e saiba onde ficam os postos de coleta. Disponível em:* <<http://www.saopaulo.sp.gov.br/ultimas-noticias/aprenda-a-descartar-o-oleo-de-cozinha-e-saiba-onde-ficam-os-postos-de-coleta/>>.  
*Acesso em: 26 outubro. 2019.*

*São Paulo, Governo do Estado. COLETA DE ÓLEO DE COZINHA. Disponível em:* <<https://www.infraestruturameioambiente.sp.gov.br/coleta-de-oleo-de-cozinha/>>.  
*Acesso em: 26 outubro. 2019.*

*Brasil Escola. TRANSFORMANDO ÓLEO DE COZINHA USADO EM SABÃO. Disponível em:* <<https://educador.brasilecola.uol.com.br/estrategias-ensino/transformando-oleo-cozinha-usado-sabao.htm/>>.  
*Acesso em: 26 outubro. 2019.*

*Óleo Sustentável. PASSO A PASSO PARA O DESCARTE ADEQUADO.*  
*Disponível em:* <<https://www.oleosustentavel.org.br/#postos-coleta>>.  
*Acesso em:* 26 outubro. 2019.

*Cidade de São Paulo Subprefeituras. Coleta Domiciliar Seletiva.*  
*Disponível em:*  
<[https://www.prefeitura.sp.gov.br/cidade/secretarias/subprefeituras/amlurb/coleta\\_a\\_seletiva/index.php?p=4623](https://www.prefeitura.sp.gov.br/cidade/secretarias/subprefeituras/amlurb/coleta_a_seletiva/index.php?p=4623)>.  
*Acesso em:* 26 outubro. 2019.

*Anne Barbosa, Bom Dia SP. Cidades da Grande SP produzem 27 mil toneladas de lixo por dia; veja para onde vão os resíduos.*  
*Disponível em:* <<https://g1.globo.com/sp/sao-paulo/noticia/2019/04/29/cidades-da-grande-sp-produzem-27-mil-toneladas-de-lixo-por-dia-veja-para-onde-vao-os-residuos.ghtml>>.  
*Acesso em:* 26 outubro. 2019.

*Portal Educação. A importância da Coleta Seletiva.*  
*Disponível em:*  
<<https://www.portaleducacao.com.br/conteudo/artigos/biologia/a-importancia-da-coleta-seletiva/61560>>.  
*Acesso em:* 26 outubro. 2019.

*Mayra Rosa. 5 motivos para plantar mais árvores.*  
*Disponível em:* <<https://ciclovivo.com.br/planeta/meio-ambiente/5-motivos-para-plantar-mais-arvores/>>.  
*Acesso em:* 26 outubro. 2019.

*Giuliana Flores. Reaproveitando Recipientes para Plantar.*  
*Disponível em:*  
<<https://blog.giulianaflores.com.br/sustentabilidade/reaproveitando-recipientes-plantar/>>.  
*Acesso em:* 26 outubro. 2019.

*Instuto Gea. Locais para entrega voluntária de materiais recicláveis.*  
*Disponível em:* <<http://www.institutogea.org.br/reciclagem/loais-para-entrega-voluntaria-de-materiais-reciclaveis/>>.  
*Acesso em:* 26 outubro. 2019.

*Pensamento Verde. Benefícios do plantio de árvores para o meio ambiente.*  
*Disponível em:* <<https://www.pensamentoverde.com.br/meio-ambiente/beneficios-plantio-arvores-meio-ambiente//>>.  
*Acesso em:* 26 outubro. 2019.

*Viva Decora PRO. Conheça as 18 árvores nativas brasileiras mais importantes (e incríveis!).*  
*Disponível em:* <<https://www.vivadecora.com.br/pro/paisagismo/arvores-nativas/>>.  
*Acesso em:* 26 outubro. 2019.