UNIVERSIDADE PAULISTA - UNIP

INSTITUTO DE ARARAQUARA

DEPARTAMENTO DE TECNOLOGIA

RICHARD WILLIAN RIBEIRO DIVINO - D07487-3
RODRIGO OLIVEIRA – D07412-1
VICTOR BARREIROS

ATIVIDADES PRÁTICAS SUPERVISIONADAS - EcoDicas

Araraquara
2019

RICHARD WILLIAN RIBEIRO DIVINO - D07487-3
RODRIGO OLIVEIRA – D07412-1
VICTOR BARREIROS

ATIVIDADES PRÁTICAS SUPERVISIONADAS - EcoDicas

Araraquara
2019

Abstract

Assim como o título, o resumo e o abstract do seu trabalho é a porta de entrada para o leitor, além de dar uma visão geral do seu trabalho, deve despertar o interesse do mesmo. Como o resumo e abstract possui uma quantidade de texto limitada, muitas pessoas tem dificuldade em elaborar um texto conciso e interessante. Desta forma, vamos apresentar uma técnica para facilitar a elaboração do resumo e o abstract que consiste em dividi-los em cinco partes: contexto, objetivo, método, resultados e conclusão.

Para mais informações acesse nosso post sobre Abstract: https://blog.fastformat.co/5-passos-resumo-e-o-abstract/

Palavras-chave: Abstract. Resumo. ABNT.

[Este é apenas um texto explicativo. Altere através do menu esquerdo.]

Lista de ilustrações

Sumário

1 Objetivos e Motivações

1.1 Geral

Desenvolver aplicação web service para conscientizar o usuário sobre a importância da proteção do eco sistema e meio ambiente além de incentivar boas práticas através de mensagens de texto. As mensagens serão enviadas diretamente em seus dispositivos android e caso o usuário perca alguma dica, a mesma estará disponível pra que ele possa visualizá-la sempre que achar necessário. A aplicação também pretende disparar informações com intervalos aleatórios aos usuários sobre fenômenos naturais de alto risco que ofereçam perigo ao usuário no cotidiano.

1.2 Específicos

Implementar o uso de servidor utilizando a linguagem de programação Ruby;

Implementar o aplicativo Eco Dicas utilizando a linguagem Flutter de programação;

Mostrar aos usuários alternativas que reduzam o impacto ambiental causado pelo consumismo e pela falta de informação.

1.3 Motivações

É um tema recorrente e importante, isso motivou o grupo a explorar o assunto em torno da sustentabilidade. A informação levada ao usuário permite que ele se conscientize e ajude na luta contra desmatamentos e desperdícios, tornando-o um forte aliado na luta pela causa.

A criação de um aplicativo nos permite alcançar uma grande quantidade de usuários, acessar vários lares e informar muitas famílias sobre a importância do consumo sustentável. Para a criação deste aplicativo o grupo utilizaria as linguagens de programação Flutter e Ruby, pois são de fácil utilização e atendia muito bem os objetivos.

2 Introdução

O desenvolvimento desordenado das cidades causado muitas vezes pela falta de emprego nas zonas rurais culmina num fenômeno conhecido como metropolização, geralmente cidades pequenas e médias tendem a sofrer mais com esse efeito. Acrescentando a isso o fato de que a maioria dos países subdesenvolvidos, com raras exceções, apresenta altas taxas de natalidade, e assim alto crescimento demográfico, formando desta forma o quadro que explica o rápido crescimento das metrópoles no mundo subdesenvolvido. O crescimento desordenado também é um fator crucial para a mudança de paisagens e florestas, desmatamento da fauna e flora da região em expansão entre outros problemas ambientais. Com o auxílio da tecnologia que está presente e cada vez mais em nosso cotidiano podemos alcançar locais onde antes seria impensável. Quanto mais rápido e mais distante ela puder se propagar maior será a quantidade de usuários informados sobre a prevenção dos riscos do desmatamento que o ser humano é capaz de causar. Nesta guerra os dispositivos móveis tornaram-se fortes aliados, isso motivou o grupo a explorar este recurso para alcançar esta população conectada oferecendo a ela lembretes dos riscos a que estão expostas, não importando sua classe social.

3 Fundamentos de Web Service e serviço de nuvem.

A API de dicas foi publicada no Heroku que é uma plataforma em nuvem como serviço (PaaS) que suporta várias linguagens de programação. Uma das primeiras plataformas em nuvem, o Heroku está em desenvolvimento desde junho de 2007, quando suportava apenas a linguagem de programação Ruby, mas agora suporta Java, Node.js, Scala, Clojure, Python, PHP e Go. Por esse motivo, o Heroku é considerado uma plataforma poliglota, pois possui recursos para um desenvolvedor criar, executar e dimensionar aplicativos de maneira semelhante na maioria dos idiomas.

Heroku foi desenvolvido inicialmente por James Lindenbaum, Adam Wiggins, e Orion Henry para apoiar projetos que eram compatíveis com a plataforma de programação Ruby conhecida como Rack. O desenvolvimento do protótipo levou cerca de seis meses. Posteriormente, a Heroku enfrentou inconvenientes devido à falta de clientes adequados no mercado, pois muitos desenvolvedores de aplicativos usaram suas próprias ferramentas e ambiente.

Os aplicativos executados no Heroku geralmente têm um domínio exclusivo (normalmente “applicationname.herokuapp.com”) usado para rotear solicitações HTTP para o dinamômetro correto. Cada um dos contêineres de aplicativos, ou dynos, estão espalhados por uma “grade dinamológica”, que consiste em vários servidores. O servidor Git do Heroku lida com o envio de repositórios de aplicativos de usuários permitidos.

Sendo assim, a plataforma Heroku, foi utilizada pela facilidade de uso e por dispor de planos gratuitos para estudantes.

4 Plano de desenvolvimento da aplicação

O desenvolvimento da aplicação tem início com a construção do servidor que centraliza as informações e mensagens, utilizando-se da linguagem de programação Ruby que estabelece a conexão entre os usuários e o servidor, em seguida realiza a troca de dados através do Flutter, ferramenta utilizada no desenvolvimento da aplicação de envio de mensagens. Inicialmente o público alvo são os usuários de dispositivos móveis, podendo expandir num segundo plano para outras plataformas.

5 Projeto (estrutura) do programa

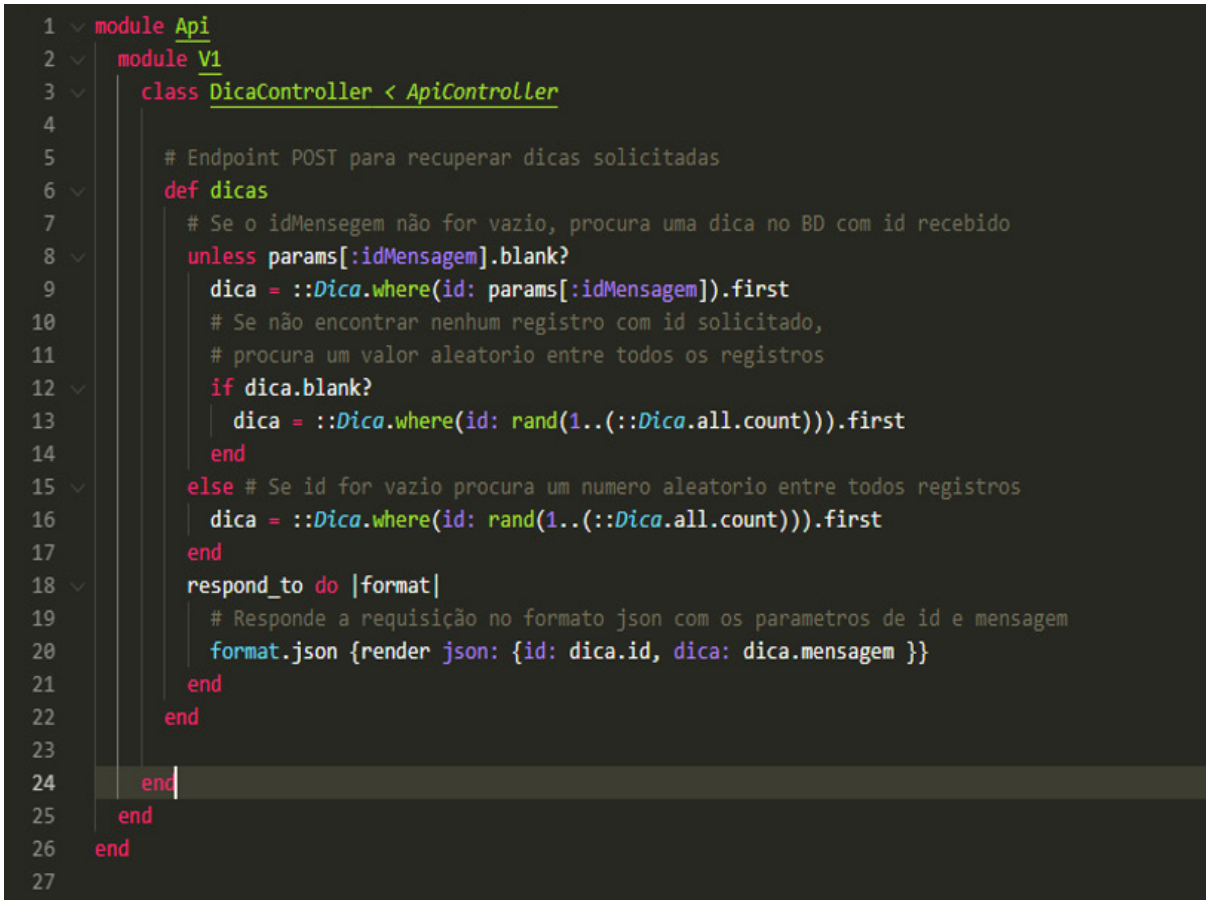
- Servidor:
  - Onde se torna possível a conexão, o envio e o recebimento dos dados, neste sistema foi criado utilizando a linguagem Ruby.
- Publicação:
  - O servidor foi publicado no Heroku, que é uma plataforma em nuvem.
- Aplicação:
  - Na aplicação foi utilizado o framework Flutter da linguagem Dart. Através dele foi possível criar um app de apresentação de mensagens tanto apara android quanto para iOS.

6 Relatório com as Linhas de código

Quando o usuário perder alguma dica, será possível revê-la.

Esta é a classe da API, responsável por receber a chamada do aplicativo, e se conectar com o banco de dados para recuperar as dicas já cadastradas.

Figura 1 – Demonstração do Controller da API, onde são feita as chamadas da aplicação



```ruby
module Api
  module V1
    class DicaController < ApiController

      # Endpoint POST para recuperar dicas solicitadas
      def dicas
        # Se o idMensegem não for vazio, procura uma dica no BD com id recebido
        unless params[:idMensagem].blank?
          dica = ::Dica.where(id: params[:idMensagem]).first
          # Se não encontrar nenhum registro com id solicitado,
          # procura um valor aleatorio entre todos os registros
          if dica.blank?
            dica = ::Dica.where(id: rand(1..(::Dica.all.count))).first
          end
        else # Se id for vazio procura um numero aleatorio entre todos registros
          dica = ::Dica.where(id: rand(1..(::Dica.all.count))).first
        end
        respond_to do |format|
          # Responde a requisição no formato json com os parametros de id e mensagem
          format.json {render json: {id: dica.id, dica: dica.mensagem }}
        end
      end

    end
  end
end

```

O Autor

Na imagem abaixo é possível demonstrar um pouco do código em Flutter que compõe a barra de navegação inferior do aplicativo:

Figura 2 – Demonstração do código de criação da Barra de Navegação Inferior

```
bottomNavigationBar: BottomNavigationBar(
  backgroundColor: Colors.white,
  iconSize: 30,
  items: const <BottomNavigationBarItem>[
    BottomNavigationBarItem(
      icon: Icon(Icons.arrow_back),
      title: Text('Dica anterior'),
    ), // BottomNavigationBarItem
    BottomNavigationBarItem(
      icon: Icon(Icons.arrow_forward),
      title: Text('Próxima dica'),
    ), // BottomNavigationBarItem
  ], // <BottomNavigationBarItem>[]
  selectedItemColor: Colors.green,
  currentIndex: _selectedIndex,
  onTap: (int index) {
    setState(() {
      _selectedIndex = index;
      _getDicas(index.toString());
    });
  },
), // BottomNavigationBar
```

O Autor

Cada BottomNavigationBarItem no código representa um botão da barra de navegação, já os outros Widgets estão relacionados as configurações de cada um destes botões e até mesmo a ações que cada botão fará, que é o caso do método onTap.

Figura 3 – Demonstração do código fonte para recuperação das dicas na API

```
15  void _getDicas(int valor) async {
16
17    var url =
18        "https://dicas-api-v1.herokuapp.com/api/dicas?idMensagem=$valor";
19
20    http.Response response;
21
22    response = await http.get(Uri.encodeFull(url),
23        headers: {"content-type": "application/json"});
24
25    var resposta = json.decode(response.body);
26
27    _dicaDia = resposta["dica"];
28    _indexDicaSelecionada = resposta["id"];
29  }
```

O Autor

O funcionamento desta etapa é demasiadamente simples, quando acionada um dos botões “Dica Anterior” ou “Próxima Dica”, é feita uma lógica na qual acrescenta ou decrementa o valor da variável _indexDicaSelecionada. O valor é enviado para a API recuperando assim a dica correspondente a ele.

Abaixo os códigos referentes ao Layout da tela principal.

Figura 4 – Demonstração de todo o código fonte referente ao *layout* da tela principal

```
return Scaffold(
  appBar: PreferredSize(
      preferredSize: Size.fromHeight(50.0),
      child: AppBar(
          backgroundColor: Colors.white,
          title: Image.asset(
            "images/EcoDicas.png",
            fit: BoxFit.cover,
            height: 1500,
          ), // Image.asset
          centerTitle: true)), // AppBar // PreferredSize
  backgroundColor: Colors.white,
  body: Stack(
    children: <Widget>[
      Image.asset(
        "images/FundoReciclagem.jpg",
        fit: BoxFit.cover,
        height: 1000.0,
      ), // Image.asset
      SingleChildScrollView(
        child: Padding(
          padding: EdgeInsets.all(10.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              Padding(
                padding:
                    EdgeInsets.only(bottom: 5, left: 20, right: 0, top: 15),
                child: Text(
                  _dicaDia,
                  textAlign: TextAlign.left,
                  style: TextStyle(
                      fontSize: 50.0,
                      color: Colors.black,
                      fontWeight: FontWeight.bold), // TextStyle
                ), // Text
              ) // Padding
            ], // <Widget>[]
          ), // Column
        ), // Padding
      ), // SingleChildScrollView
```

O Autor

Explicando alguns Widgets do código:

- AppBar: Representa a barra superior onde está localizado o logo da aplicação EcoDicas;
- Stack: Este Widget é considerado um construtor, pois dentro dele são adicionados outros Widgets que comporão a estrutura final. Ele é utilizado para que Widgets possam sobrepor outros Widgets. No caso da aplicação em questão,

as dicas estão acima da imagem de fundo e não há uma concorrência devido a composição do Stack;

- SingleChildScroolView: É um Widget responsável pela criação de uma barra de rolagem caso os elementos da tela ultrapassem do tamanho total dela.

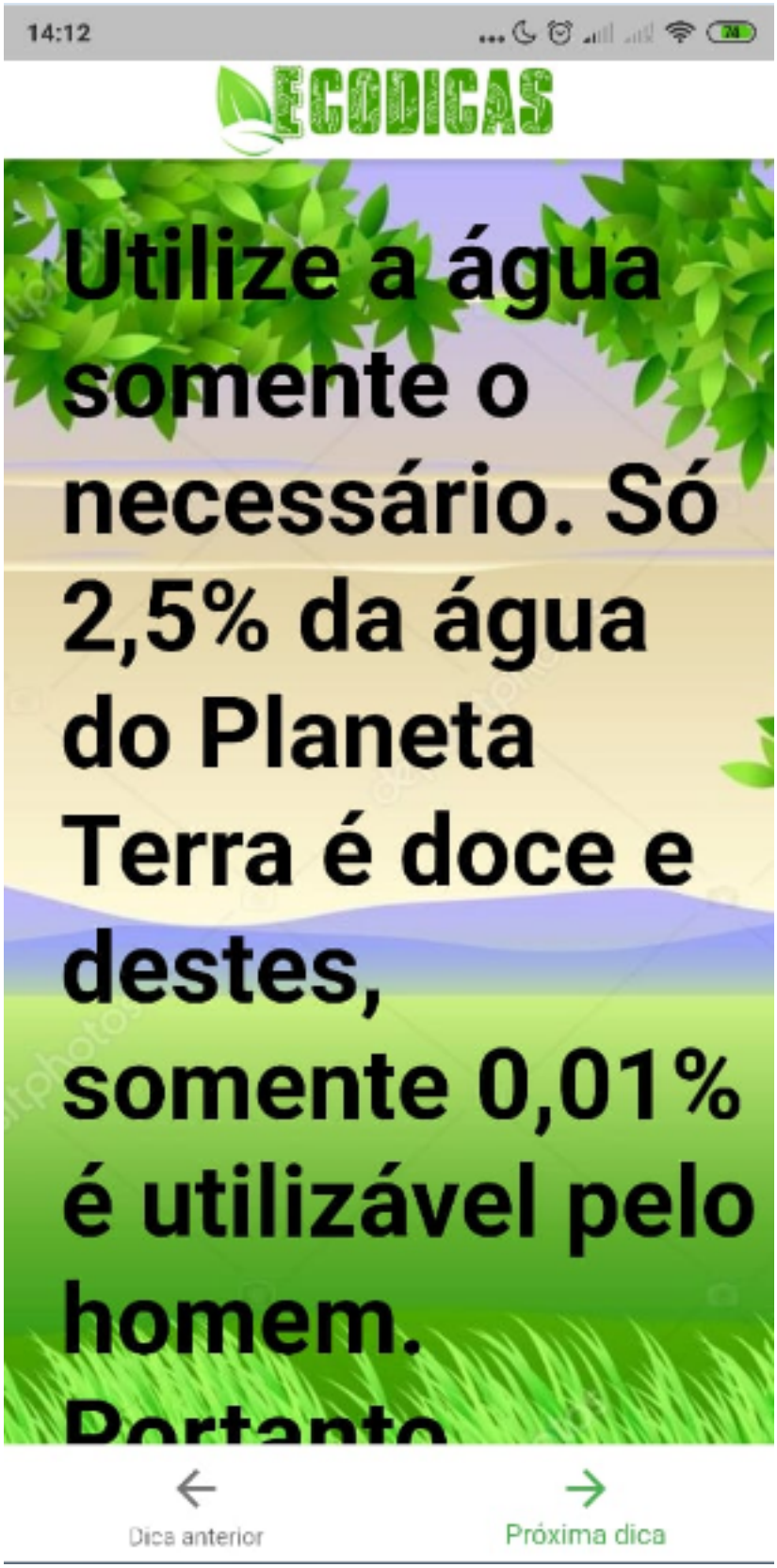
Os demais *Widgets* como *EdgeInsets, Padding, FontWeight,* etc. São responsáveis pela customização visual da aplicação, existem muitos outros deste tipo, porém a aplicação é simples e não demandava tanto código.

7 Apresentação do funcionamento do aplicativo

Não será necessária nenhuma configuração por parte do usuário para que ele receba as informações da aplicação, ao iniciar o aplicativo em seu dispositivo móvel as mensagens são apresentadas automaticamente, após exibir todas as mensagens o sistema carrega aleatoriamente uma das mensagens e o loop de imagens continua.

A imagem abaixo exemplifica como será exibida a mensagem da aplicação no celular do usuário.

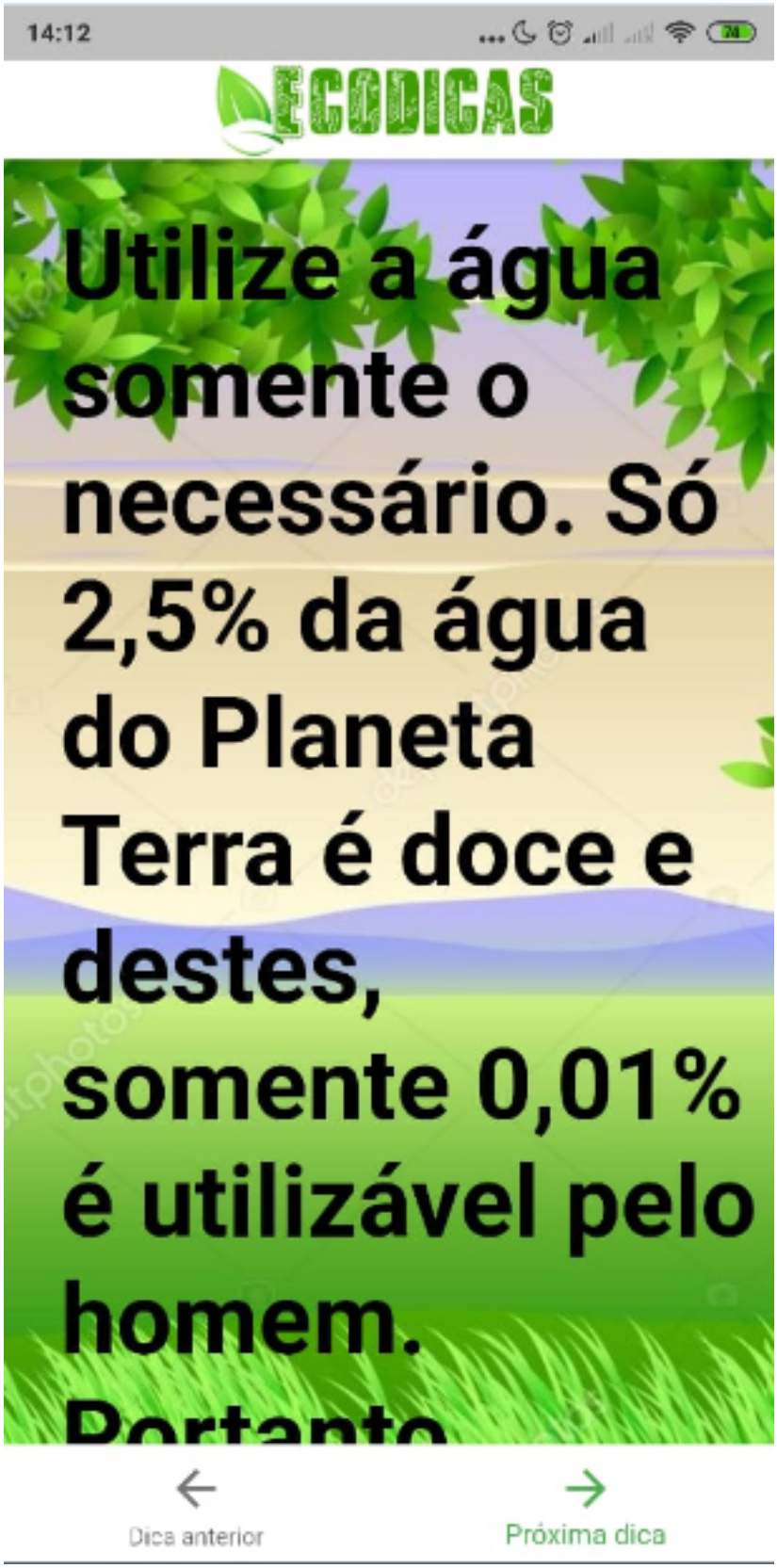
Figura 5 – Exemplo 1 de dica apresentada no aplicativo



O Autor

Ao clicar no botão “Próxima Dica”, o aplicativo buscará a próxima dica do servidor

Figura 6 – Exemplo 2 de dica apresentada no aplicativo



O Autor

Caso o usuário venha a clicar no botão “Dica Anterior” ele retornará ao exemplo

1 de estado da aplicação.

Serão ao todo 10 mensagens relacionadas ao reuso e preservação do meio ambiente.

1) Economizar água é algo essencial;

2) Evite o desperdício de papel e contribua para a redução do corte de árvores e do lançamento dos gases que formam o efeito estufa. Use o outro lado dos papéis como rascunho;

3) Utilize a água somente o necessário. Só 2,5% da água do Planeta Terra é doce e destes, somente 0,01% é utilizável pelo homem. Portanto, nunca lave uma calçada utilizando uma mangueira. Use um balde;

4) Deixe o carro na garagem e utilize o transporte coletivo e a bicicleta, quando possível. Dê preferência a combustíveis como o álcool e o biodiesel. Faça revisões periódicas no seu veículo para reduzir as emissões de poluentes;

5) Ajude a recuperar o verde de sua cidade. Plante árvores no seu quintal, na sua propriedade rural e até mesmo em áreas públicas;

6) Evite o desperdício de água. Feche sempre a torneira quando não estiver em uso. Em áreas sujeitas a secas prolongadas, armazene água. E arrume as torneiras que estão pingando. Uma torneira pingando desperdiça mais de 40 litros de água por dia;

7) Economize energia. Troque lâmpadas incandescentes por fluorescentes, apague luzes desnecessárias, desligue aparelhos domésticos quando não estiverem em uso e compre eletrodomésticos classificados como nível A em eficiência energética;

8) Não compre mais que o necessário, pois certamente vai virar lixo dentro da sua casa, ocupando espaço, muitas vezes, por anos a fio e sem utilidade, portanto, se tem excessos, faça trocas ou doações;

9) Desligue o computador. Muita gente tem o péssimo hábito de deixar o computador ligado ininterruptamente, às vezes fazendo downloads, às vezes por pura comodidade;

10) Use sacolas de pano em vez das de plástico. Sacolas de plástico – quando jogadas nas vias públicas – entopem os esgotos e provocam enchentes.