



**Curso de Ciência da Computação**

**Atividades Práticas Supervisionadas - APS**

**DESENVOLVIMENTO DE SOFTWARE UTILIZANDO CONCEITOS DE  
PROGRAMAÇÃO ORIENTADA A OBJETOS**

**Guilherme Augusto Sbizero Correa RA: F235289**

São Paulo  
2021



## RESUMO

O trabalho tem como objetivo principal o desenvolvimento de um software utilizando a Linguagem Orientada a Objetos para criar algo relacionado com o meio ambiente e sua importância, facilitando o entendimento do usuário para que tenha uma visão ampla sobre o assunto a ser programado e possa compartilhar as informações obtidas para colegas e familiares, levando o conhecimento obtido para ser utilizado dentro de casa, entre quaisquer outros lugares.

No início é possível notar que muitas pessoas não têm conhecimento do que é possível fazer quando uma planta está morrendo, um arbusto está desidratado, quando uma planta está diferente do que era previsto, não sabe lidar com essas e várias outras situações. O aplicativo que foi desenvolvido tem como principal objetivo trazer informações para que o usuário possa cuidar de suas plantas e flores que deseja ter em sua casa, mas não sabe como lidar com o que é necessário para ter.

O meio ambiente é um termo principal em nosso planeta terra, que pode tanto salvá-lo como também pode destruí-lo se não for administrado, então é necessário ter um aprendizado, pois até mesmo uma planta que mal é notada, pode fazer uma grande diferença, portanto foi criado um aplicativo para que cada usuário saiba o que fazer e que tenha um jardim do jeito que desejar.

O software que foi desenvolvido na linguagem de programação JAVA, a linguagem mais utilizada pelos programadores e a melhor até o presente momento, sendo um pouco complicada de ser utilizada, mas com esforço e dedicação foi possível fazer um aplicativo na qual o usuário possa entender e levar para sua vida, compartilhando o que foi aprendido e possibilitando um crescimento no nosso projeto.

A estratégia é fazer um programa de fácil entendimento para que ele mesmo faça seu próprio marketing e que leve o que é ensinado para o maior número

possível de pessoas, pois a dificuldade em cultivar o que é desejado existe, mas a solução para esses problemas é difícil de ser encontrada, então esperamos que o trabalho feito com responsabilidade e esforço seja recompensado com o principal, a transição do conhecimento adquirido e reconhecimento pelos nossos serviços.



## **Objetivo do trabalho**

O trabalho tem como principal objetivo trazer para os usuários uma nova experiência com o aplicativo, na qual em forma de aprendizado ele possa realizar todo o passo a passo e conseguir ter um jardim da forma desejada, portanto é desejado mostrar essa nova perspectiva para o nosso usuário, é algo diferenciado e inovador, então fica mais difícil para que seja reconhecido, por isso contamos com a iniciativa dos mesmos para que façam o marketing, mostrando para amigos e conhecidos, para que todos saibam que é possível ter um jardim organizado, limpo e bem tratado com muito menos tempo e com grande facilidade.

E normal que quando se tem o interesse de ter um jardim, acaba tendo um desinteresse por ter um processo gigantesco para iniciá-lo, pois tem que ter tempo para comprar todos os acessórios que são necessários, e também as outras coisas para o manter, então quando se pensa em toda essa burocracia acabam desistindo de fazer e preferem deixar como está, pois é considerado uma perda de tempo

Por isso esse aplicativo é importante, ele vai facilitar a vida do usuário para que ele se incentive a criar o seu jardim e tenha todo o processo na palma da sua mão, com informações acrescentadas, dicas de como mantê-los e localidade necessária, para que não haja problemas futuros, exemplos de jardins e pessoas que ajudaram para um mundo mais limpo, com um foco para um meio ambiente preservado e um espaço bem utilizado e administrado.

O mais importante é trazer a atenção ao meio ambiente para que todos notem que essa parte do nosso planeta tem que ser cuidada e bem utilizada, dividindo interesse entre ambos, as pessoas e ao meio ambiente. É um assunto que deve ser lembrado e discutido, e que todas as pessoas tenham informações sobre esta área, abrangendo o conhecimento de todos e sendo avisadas

diariamente sobre o que é alterado e o que deve ser feito em todas as ocasiões necessárias.

Enfim, é o que esse aplicativo está tentando promover, trazer o usuário para mais perto do meio ambiente dando informações para seu jardim, mas não apenas sobre isso, também é possível trazer sobre diversos assuntos sobre este tema principal, é um começo, mas ainda não é visto um final.

## Sumário

1. Introdução .....	13
2. Elementos de programação orientada a objetos. ....	15
2.1 Pilares da POO.....	15
2.2 Abstração .....	15
....2.3 Encapsulamento.....	15
....2.4 Herança.....	15
....2.5 Polimorfismo.....	15
....2.6 Principais vantagens da POO.....	15
....2.7 Exemplos de linguagens Orientadas a Objetos .....	15
3. Descrição dos módulos do sistema.....	21
4. Projeto do sistema.....	26
5. Bibliografia.....	32



## 1.0 INTRODUÇÃO

Através deste trabalho, iremos apresentar ao leitor uma pesquisa abordando o tema educação ambiental. Onde nela conterá informações sobre os principais tópicos da educação ambiental ligadas ao meio ambiente como, aquecimento global e as mudanças climáticas, a poluição de águas rios e oceanos e os danos provocados por fontes de energias não renováveis. Essa pesquisa conscientizará o usuário de como é importante cuidar do meio ambiente, a falta de conscientização sobre o meio ambiente, causa não só terremotos, tsunamis e extinção de animais, como também interferem na qualidade de nossas vidas e de gerações futuras.

Com base nessa pesquisa, na parte prática será desenvolvido um software orientado a objetos, usando a linguagem de programação Java, o Programa conterá informações educacionais sobre o meio ambiente, com o objetivo de informar o usuário a importância de ter conhecimento sobre o tema meio ambiente. O software se baseia em 3 áreas, na qual ajudam o usuário a criar um jardim ou uma horta, como o usuário pode contribuir para o meio ambiente, e por fim, mostrar 3 grandes famosos contribuintes do meio ambiente.

O software respeitará os princípios da orientação a objetos (classes, objetos, heranças e polimorfismo). O aplicativo conterá comentários em seu código fonte, a fim de ajudar no entendimento de como foi produzido toda lógica envolvente no programa.

Já na parte teórica, com o objetivo de fazer com que o leitor entenda como o software (Aplicação) foi desenvolvido pelos programadores do nosso grupo, será apresentado ao leitor todo conceito da linguagem orientada a objetos, onde veremos os quatro pilares desse paradigma e entendemos como eles são implementados em algumas das linguagens mais utilizadas no mercado de desenvolvimento. Além disso, será apresentado algumas das vantagens que

tornaram a programação orientada a objetos um grande sucesso para o desenvolvimento de sistemas modernos, tudo isso para que o leitor entenda como o software foi desenvolvido através da pesquisa. O leitor poderá não só entender como o software foi produzido, como toda a parte conceitual que envolva a programação de software em geral, isso estimula o interesse por programação e dá uma compreensão e perspectiva de como um software é desenvolvido pelos programadores.

As telas de explicação do software serão criadas com o intuito para que o usuário do aplicativo possa usar o software com clareza e tenha conhecimento das informações que nele existe

As linhas de códigos que compõe a aplicação criada abordando o tema “educação ambiental” será disponibilizada para que o leitor tenha conhecimento da parte não física da aplicação em si.

Junto com as linhas de códigos, comentários feitos pelos programadores do software, serão implementados no trabalho com o intuito de mostrar, ensinar e fazer com que o leitor entenda o que está sendo feito em cada linha de código, cada passo a passo que compõe toda a programação feita. Tudo isso para que ele entenda os dois lados da história da programação, desde a criação do software utilizando uma linguagem de programação orientada a objetos até a execução desta aplicação.

## **Elementos de programação orientada a objetos**

De acordo com GASPAROTTO (2014), no conceito de linguagens de programação, a P.O.O (Programação orientada a objetos) é o paradigma mais difundido entre todos existentes no mercado. Isso porque o seu padrão para desenvolvimento de softwares tem ganhado cada vez mais evolução, uma característica que pode ser citada é a do reaproveitamento do código desenvolvido que nesse caso pode ser considerado o mais importante, caracterizando como uma aplicação mais moderna, segura e facilidade de desenvolvimento.

Um paradigma que se baseia em quatro pilares, diferente da linguagem estruturada, que era muito utilizada há alguns anos. A diferença entre as duas linguagens começa na base de seus códigos e aplicações. No paradigma estruturado temos o que chamamos de procedimentos “funções” seguimentos do meu programa a ser desenvolvido, já na P.O.O, temos aplicações aos dados de cada objeto, característica denominada como “Métodos”. Procedimentos e métodos são semelhantes, sendo diferenciados apenas pelo escopo do programa (contexto delimitante aos quais valores e expressões estão associados).

A programação orientada a objetos apresenta para o programador pontos mais interessantes do que a linguagem estruturada, pelo fato de ser uma linguagem mais moderna, o desenvolvimento de softwares não se torna uma grande preocupação para o desenvolvedor, devido a característica de que pode ser processada nos computadores atuais, pela reutilização do código já desenvolvido em outras aplicações e a proximidade com o mundo real representado pela apresentação do sistema já pronto. GASPAROTTO, 2014)

## **Pilares da POO**

Para uma linguagem ser caracterizada como orientação a objetos, ela deve atender quatro requerimentos, são eles (Abstração, encapsulamento, herança e polimorfismo). (GASPAROTTO, 2014).

### **Abstração**

Pontos importantes dentro de qualquer linguagem orientada a objetos na POO lidaram com uma representação de um objeto real e temos que imaginar as funções que esse mesmo objeto realizara dentro do nosso sistema. Devem ser levadas em consideração três características para a abstração.

O primeiro é darmos uma identidade para o objeto a ser criado. A identidade é uma propriedade que permite identificar univocamente um objeto, os objetos se distinguem por sua própria existência. Nessa linguagem não pode haver identidades referenciadas ao objeto repetidas dentro dos pacotes (classes) e não necessariamente em todo o sistema.

A característica desse objeto se dá a segunda parte, um objeto possui características que facilitam sua identificação, na POO essas características são denominadas como “propriedades” do objeto. Para melhor entender, um objeto carro tem como características (propriedades) “cor, modelo”.

E a terceira característica da abstração é definir as ações que o objeto criado irá realizar em nosso sistema, chamamos essa ação de métodos. Usando o exemplo de o carro andar podemos definir um método “acelerar” como sua ação. (GASPAROTTO, 2014).

## **Encapsulamento**

Pode ser considerada a característica mais importante dentro da programação orientada a objetos, um pacote contendo objetos deve ser desenvolvido de tal forma que somente seu comportamento seja conhecido pelo sistema, e não sua implementação. Elemento que esconde as propriedades do objeto e que dá segurança a minha aplicação, como se fosse uma caixa preta de um avião. Essa implementação evita acesso direto a informações do meu objeto através de outra camada de segurança a aplicação. Ainda usando o exemplo do objeto carro, vamos usar o método “ligar” para dar uma ação ao objeto, ao ligarmos o carro não sabemos os processos que levam do girar a chave no contato até a ligação do motor, apenas entendemos que o carro foi ligado. GASPAROTTO, 2014).

## **Herança**

Como já dito, a reutilização de códigos é uma das principais características presente da programação orientada a objetos, essa questão se dá ao nome “herança”. Uma característica que aperfeiçoa tempo de produção da aplicação e das linhas de códigos. Pode ser citado como um exemplo claro uma “família” qualquer, onde filhos herdaram características de seus pais e seus pais herdaram características de seus avós. GASPAROTTO, 2014).

## **Polimorfismo**

Tem relação direta com minhas classes e objetos, neste caso meu objeto criado tem todos os seus métodos citados, usando a herança para outro objeto, vamos ter os mesmos métodos, porém alguns deles terão características diferentes então eles deverão ser reescritos. Vamos usar “eletrodoméstico”

como um objeto, o mesmo possui o método “ligar”, dentro desse pacote temos dois objetos diferentes “geladeira” e “televisão” sendo da mesma classe eletrodoméstico, tendo o método ligar ainda como ação, sabe que elas não serão ligadas da mesma forma, então entendemos que o método “ligar” herdado dos dois objetos deverão ser escritos de forma diferente.

Tendo conhecimento sobre os esses quatro pilares essenciais, podemos entender facilmente a linguagem de programação orientada a objetos. (GASPAROTTO, 2014).

### **Principais vantagens da POO**

Esse tipo de linguagem tenta ao máximo aproximar o sistema desenvolvido do que podemos observar do mundo real, através da representação de um objeto ou uma classe. Como as ações “métodos” implantados a um objeto “classe” como vemos na realidade, essa representação traz vantagens para o programador que procura criar uma aplicação e também para o usuário dessa mesma aplicação.

Podemos dizer que a criação de uma aplicação comum é bem complexa, o que levaria tempo até o final do seu desenvolvimento, mas através da reutilização do código fonte da minha programação orientada a objetos, esse tempo de desenvolvimento é diminuído, junto com o número total de linhas de códigos, isso só é possível devido a POO trazer representações claras de cada um de seus elementos.

A manutenção e leitura do código também é uma outra grande vantagem da POO, diferente da linguagem estruturada, onde só o desenvolvedor do código teria conhecimento do sistema, apenas o mesmo seria capaz de interpretar erros do sistema e assim fazer a manutenção do mesmo, a programação orientada a objetos que está próxima a nossa realidade permite que a leitura e manutenção dos códigos fontes seja feito separadamente, ou seja podem ser feitos por duas ou mais pessoas da equipe de desenvolvimento do sistema. A manutenção e o impacto causados por alterações em requisitos e novas regras de negócio ficam bastante minimizados pelo uso de tecnologia de objetos.

No entanto, nem tudo pode ser considerado perfeito na programação orientada a objetos. Diferente da linguagem estruturada, uma aplicação desenvolvida na POO ao ser executada, não ocorre com a mesma rapidez, se torna mais lenta devido à complexidade de desenvolvimento do modelo, representadas em formas de classes, nelas são encontrados os desvios que causam a lentidão não tão brusca da aplicação, já na estruturada ocorre de forma sequencial (linguagem “C”).

Mas na nossa realidade, tratando da tecnologia de linguagem de programação a execução mais lenta não é praticamente sentida. O que faz com que a programação orientada a objetos seja a mais recomendada pelos desenvolvedores de sistemas devidos as vantagens citadas e apresentadas, baseadas em representações que vimos no mundo real. GASPAROTTO, 2014).

### **Exemplos de linguagens orientadas a objetos**

Existem diversas linguagens de programação orientada a objetos no mercado. Nesta pesquisa iremos apresentar três que são consideradas as mais utilizadas pelos desenvolvedores de sistemas. São elas (Java; C#; C++), essas linguagens possuem abordagens diferentes para aplicações caracterizando umas boas para certas aplicações e ruins para outras representações. GASPAROTTO, 2014).

#### **Java**

Lançado em 1991 provavelmente estamos falando da linguagem mais utilizada para desenvolver aplicações na programação orientada a objetos. Presente em quase todos dispositivos eletrônicos, é sucesso entre desenvolvedores, sua utilização aumentou ainda mais quando o Google android, escolheu o Java como linguagem preferencial para desenvolver seus aplicativos.

Atende todos os requisitos que implementam os 4 pilares da programação orientada a objetos, além de ter a condição da portabilidade do seu código fonte, isso que significa que você pode desenvolver as aplicações no sistema operacional Windows e executá-los no sistema operacional Linux, sem necessidade de reescrever o código. GASPAROTTO, 2014).

## **C#**

Outra linguagem bem utilizada pelos desenvolvedores popularizou-se através dos computadores quem em sua maioria tem como sistema padrão o Windows da rede Microsoft, é associado a uma linguagem de uso geral e especialmente a de programação orientada a objetos (seguindo todos os pilares do paradigma de linguagem GASPAROTTO, 2014).

## **C++**

Diferente das duas outras citadas e das demais, essa linguagem é um pouco mais primitiva, e permite ter mais liberdade com os hardwares. Derivada da linguagem C, trabalham com ponteiros ligados diretamente com a memória do sistema, podendo também utilizar todas as bibliotecas que compõe a linguagem “c”.

Brevemente podemos citar outras linguagens de programação orientadas a objetos e suas principais aplicações, como a “Python” que é uma linguagem do tipo script que é muito utilizada em pesquisas científicas devido sua velocidade de execução “Pascal” também conhecida como “Delph”, e a principal linguagem para desenvolver aplicações para a apple voltada para desenvolvimento web, e a “Visual Basic” linguagem muito utilizada até pouco tempo, que vem em desuso, devido ao avanço principal da linguagem “C” por sua popularidade. GASPAROTTO, 2014).



## Descrição dos módulos do sistema

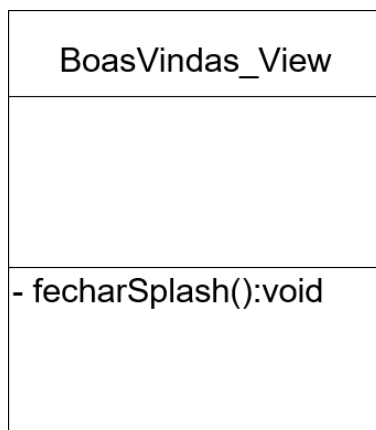


Figura 1- Classe BoasVindas\_View . Fonte: próprio autor

**Método fecharSplash:** Esse método é responsável por fechar a tela inicial do software e direcionar a próxima tela, a principal do programa.

Na figura 2 é representada a tela do software, onde há três opções para o usuário: “Meio ambiente na sua casa”, “Sua ajuda ao meio ambiente” e “Heróis do meio ambiente”.

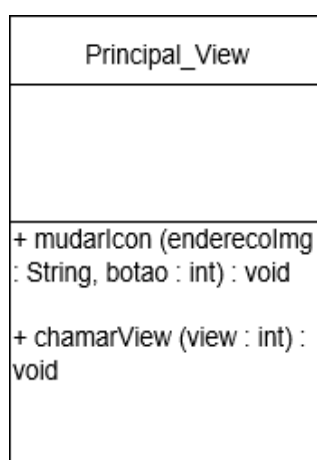


Figura 2- Classe Principal\_View . Fonte: próprio autor

**Método mudarIcone:** Esse método é responsável pela mudança de cores nos botões e ícones, o mesmo recebe um endereço de imagem com o efeito em si, e recebe a identificação do botão para a criação do efeito.

**Método chamarView:** Esse método é responsável pela chamada de uma view (tela). Ele recebe um inteiro como identificação da view, e direciona para a view de acordo com a identificação.

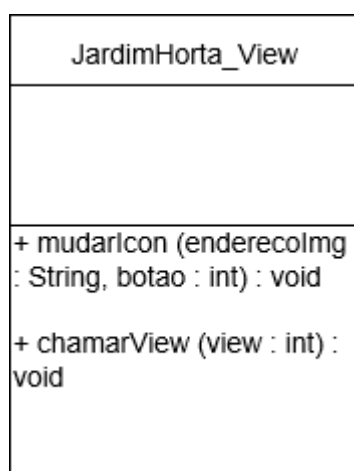


Figura 3- Classe JardimHorta\_View. Fonte: próprio autor

**Método mudarIcone:** Esse método é responsável pela mudança de cores nos botões e ícones, o mesmo recebe um endereço de imagem com o efeito em si, e recebe a identificação do botão para a criação do efeito.

**Método chamarView:** Esse método é responsável pela chamada de uma view (tela). Ele recebe um inteiro como identificação da view, e direciona para a view de acordo com a identificação.

Jardim_View
- paginasJardim : int - paginasPlantas : int
+ mudarIcon (enderecolmg : String, botao : int) : void  + chamarView (view : int) : void  + inicializarComponentes () : void

Figura 4- Classe Jardim\_View. Fonte: próprio autor

**Método mudarIcon:** Esse método é responsável pela mudança de cores nos botões e ícones, o mesmo recebe um endereço de imagem com o efeito em si, e recebe a identificação do botão para a criação do efeito.

**Método chamarView:** Esse método é responsável pela chamada de uma view (tela). Ele recebe um inteiro como identificação da view, e direciona para a view de acordo com a identificação.

Horta_View
+ mudarIcon (enderecolmg : String, botao : int) : void  + chamarView (view : int) : void  + inicializarComponentes () : void

Figura 5- Classe Jardim\_View. Fonte: próprio autor

**Método mudarIcone:** Esse método é responsável pela mudança de cores nos botões e ícones, o mesmo recebe um endereço de imagem com o efeito em si, e recebe a identificação do botão para a criação do efeito.

**Método chamarView:** Esse método é responsável pela chamada de uma view (tela). Ele recebe um inteiro como identificação da view, e direciona para a view de acordo com a identificação.

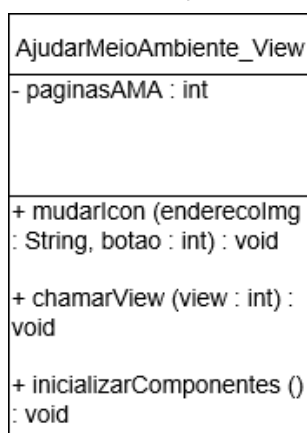


Figura 6– Classe AjudarMeioAmbiente\_View. Fonte: próprio autor

Essa classe contém um método para inicializar os componentes de maneira personalizada, e esse método modifica a visibilidade dos botões “avançar página” e “voltar página”, chamado de **inicializarComponentes**.

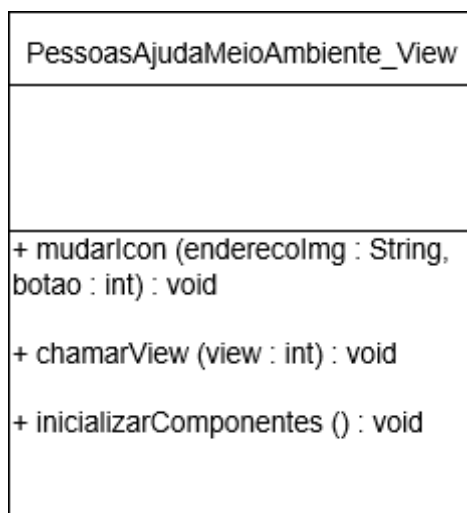


Figura 7– Classe PessoasMeioAmbiente\_View. Fonte: próprio autor

**Método mudarIcon:** Esse método é responsável pela mudança de cores nos botões e ícones, o mesmo recebe um endereço de imagem com o efeito em si, e recebe a identificação do botão para a criação do efeito.

**Método chamarView:** Esse método é responsável pela chamada de uma view (tela). Ele recebe um inteiro como identificação da view, e direciona para a view de acordo com a identificação.

**inicializarComponentes.**Essa classe contém um método para inicializar os componentes de maneira personalizada, e esse método modifica a visibilidade dos botões “avançar página” e “voltar página”.

## Projeto (Estrutura do programa)

A figura 1 representa uma splash screen da tela inicial do software de conscientização do meio ambiente:



Figura 1- Tela inicial do software FF Ambiental (Fauna e Flora Ambiental).  
Fonte: próprio autor.

Na figura 2 é representada a tela do software, onde há três opções para o usuário: “Meio ambiente na sua casa”, “Sua ajuda ao meio ambiente” e “Heróis do meio ambiente”.



Figura 2- Tela principal do software. Fonte: próprio autor.

As figuras 3, 4 e 5 fornecem ao usuário duas opções: aprender a cuidar de um jardim, ou de uma horta. Dentro de cada opção, aparecerão informações de acordo com o que o usuário escolher.



Figura 3- Tela resultante de “Meio ambiente na sua casa”. Fonte: próprio autor.

O jardim e a horta recebem os mesmos métodos da tela principal: “mudarIcone” e “chamarView”.



As duas imagens a seguir são resultado do que o usuário escolher entre jardim ou horta e as informações serão repassadas para o mesmo de acordo com sua escolha.

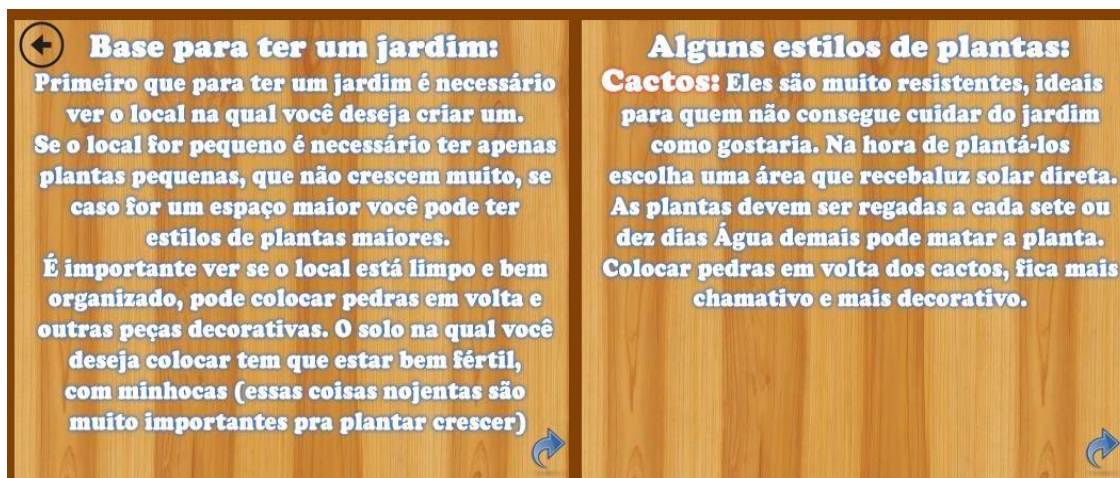


Figura 4- Informações sobre como criar e cuidar de um jardim. Fonte: próprio autor.

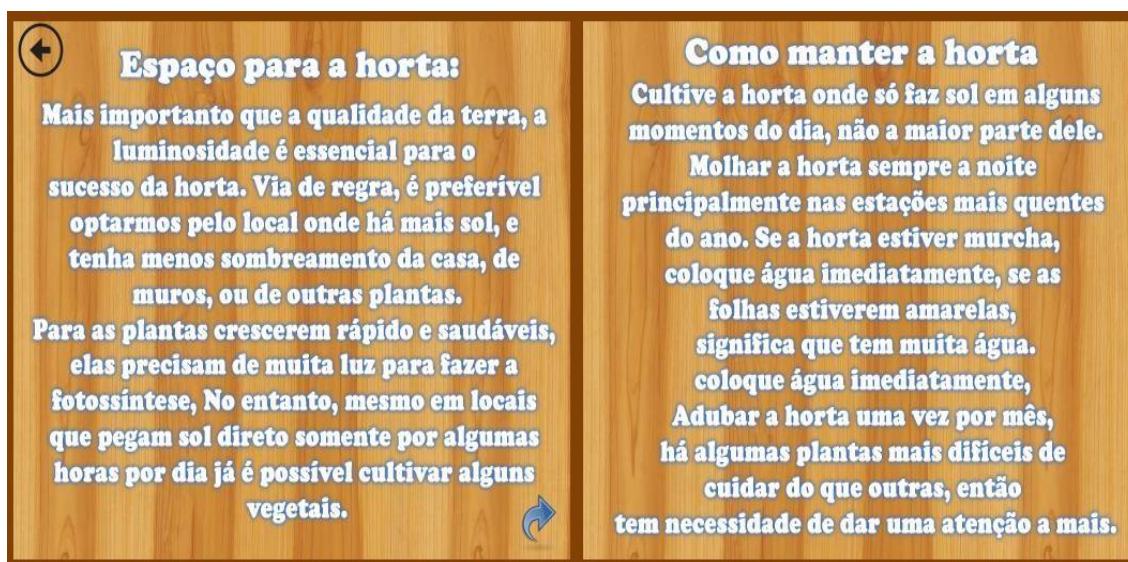


Figura 5- Informações sobre como criar e cuidar de uma horta. Fonte: próprio autor.



Ao obter todas as informações sobre como criar e cuidar de uma horta ou de um jardim, o usuário poderá voltar à tela principal do software. O próximo tópico a ser abordado, na figura 6, é a sua ajuda ao meio ambiente, onde se encontram as dicas para a boa utilização do mesmo.



Figura 6- Tela que aborda as dicas para ajudar o meio ambiente. Fonte: próprio autor.

A tela de dicas chama os mesmos métodos anteriormente citados: **chamarView** e **mudarIcone**. Essa tela contém um método para inicializar os componentes de maneira personalizada, e esse método modifica a visibilidade dos botões “avançar página” e “voltar página”, chamado de **inicializarComponentes**.

Ao todo são 10 dicas apresentadas ao usuário do software, e por motivos de conveniência apenas uma é apresentada no trabalho, a figura 7.

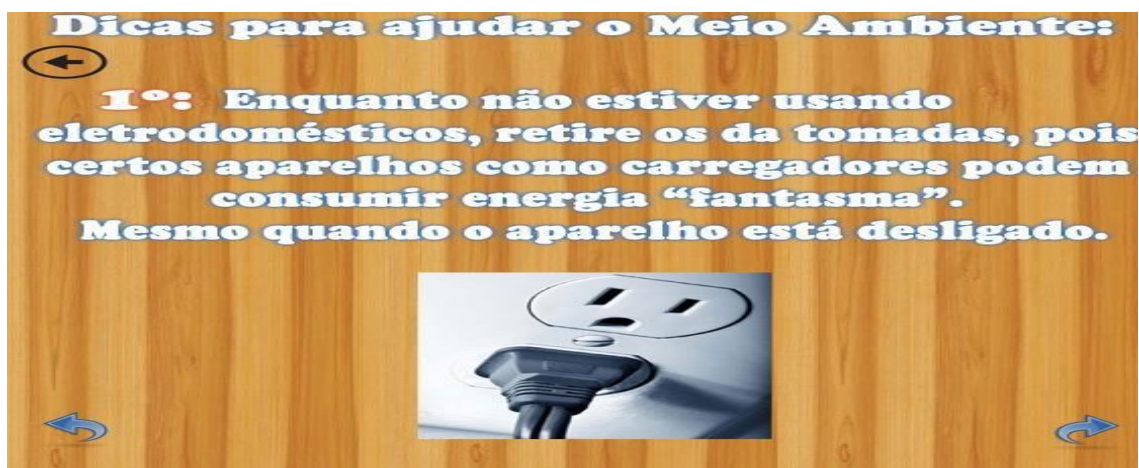


Figura 7- Primeira dica apresentada ao usuário ao clicar em “Sua ajuda ao meio ambiente”. Fonte: próprio autor.

Ao escolher a opção de “Heróis do meio ambiente”, o usuário será informado na figura 8 sobre três celebridades mundiais: Leonardo DiCaprio, Gisele Bündchen e Angelina Jolie, todos com suas respectivas ações importantes para a preservação do meio ambiente. Agora basta ao usuário escolher sobre

quais celebridades deseja obter informações.



Figura 8- Celebidades que ajudaram o meio ambiente de alguma forma. Fonte: próprio autor.

A figura 9 é destinada a mostrar as informações sobre os feitos dos contribuintes do meio ambiente, neste exemplo, DiCaprio.

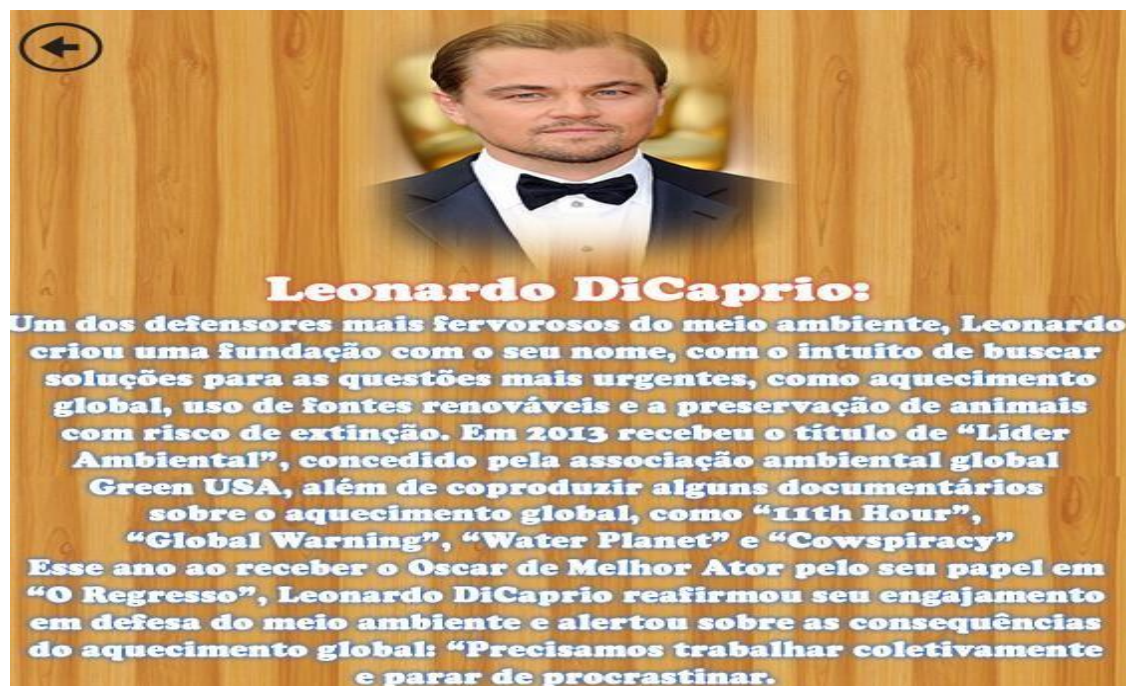


Figura 9- Leonardo DiCaprio é apresentado na tela de “Heróis do meio ambiente”. Fonte: próprio autor.

## BIBLIOGRAFIA

GASPAROTO, Henrique Machado. Elementos da linguagem de programação. Os 4 pilares da programação orientada a objetos: 2014  
Disponível: <https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>  
Acesso dia 20/04/2019

SANTOS, Marco Aurélio dos. Energia. Meio ambiente: LTC 2014  
Disponível: [https://www.suapesquisa.com/energia/fontes\\_ao\\_renovaveis.htm](https://www.suapesquisa.com/energia/fontes_ao_renovaveis.htm)  
Acesso dia: 20/04/2019

FORUM DAS ONGS. Meio ambiente e Desenvolvimento: Uma visão das ONG'S e dos Movimentos Sociais Brasileiros. Rio de Janeiro, 1992  
Disponível: [https://pt.wikipedia.org/wiki/Educa%C3%A7%C3%A3o\\_ambiental](https://pt.wikipedia.org/wiki/Educa%C3%A7%C3%A3o_ambiental)  
Acesso dia 20/04/2019

REVISTA GAIA SCIENTIA. Mudanças climáticas e conservação. Risco do aquecimento global: ISSN, 1981  
Disponível: <http://periodicos.ufpb.br/index.php/gaia/article/view/3342>  
Acesso dia 20/04/2019

UMULTRAMOLECULAS. Meio Ambiente. Como podemos evitar a poluição das águas, 2018  
Disponível: <https://g1.globo.com/sp/itapetininga-regiao/especial-publicitario/um/troque-todos-por-um/noticia/2018/08/23/como-podemos-evitar-a-poluicao-das-aguas.ghtml>  
Acesso dia 20/04/2019

[illegible]













