**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**HENRIQUE LEAL TAVARES**

**APLICATIVO HÍBRIDO INTEGRADO À PLACA EMBARCADA PARA GERENCIAMENTO DE UM ESTACIONAMENTO**

**GARÇA**

**2017**

**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**HENRIQUE LEAL TAVARES**

**APLICATIVO HÍBRIDO INTEGRADO À PLACA EMBARCADA PARA GERENCIAMENTO DE UM ESTACIONAMENTO**

Relatório técnico e/ou científico apresentado à Faculdade de Tecnologia de Garça – Fatec, como requisito para conclusão do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador: Profº. Ms. João Baptista Cardia Neto

**GARÇA**

**2017**

L. Tavares, Henrique

Aplicativo híbrido integrado à placa embarcada para gerenciamento de um estacionamento/ Henrique Leal Tavares. - Garça, 2017.

31 f.

Monografia (Trabalho de Conclusão de Curso de Tecnologia em Analise e Desenvolvimento de Sistemas). – Faculdade de Tecnologia de Garça – FATEC, 2017.

Orientador: Prof.Ms. João Baptista Cardia Neto

1. TCC. 2.Controle de Atividades. 3. Projetos. I. Autor. II. Título.

CDD 004.35

**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**HENRIQUE LEAL TAVARES**

**APLICATIVO HÍBRIDO INTEGRADO À PLACA EMBARCADA PARA GERENCIAMENTO DE UM ESTACIONAMENTO**

Relatório técnico e/ou científico apresentado à Faculdade de Tecnologia de Garça – Fatec, como requisito para conclusão do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, examinado pelos professores:

Profº. Ms. João Baptista Cardia Neto

FATEC Garça

Prof.

FATEC Garça

Prof.

FATEC Garça

Data da Aprovação:

**GARÇA**

**2017**

**RESUMO**

O gerenciamento de estacionamento pode se tornar complexo, principalmente quando o estacionamento pertence a um local que deve prover acesso a todos, como uma faculdade. Essa dificuldade se referente a validação de acessos ao mesmo quanto a cadastros dos veículos, funcionários e relatórios que poderiam ser gerados a partir destas informações. Pesando nesse sentido é necessário propor um aplicativo híbrido com comunicação expressa com uma placa embarcada para a administração de todos os dados e automação do controle das cancelas do estacionamento. O *software* deverá realizar todos os cadastros e validações de entrada e saída automaticamente, mas também será capaz de permitir que um administrador faça todos os processos manualmente caso ocorra quedas de energias ou falhas com os equipamentos implantados para o funcionamento da cancela. Foi utilizado no desenvolvimento do *front-end* o *framework* Ionic 2 que é um *framework* que utiliza componentes criados em HTML5 e CSS3 já moldados para o cenário mobile, por sua vez no *back-end* o *framework* AngularJS, um *framework* mantido pela Google que consegue ser compilado em diversas plataformas, sejam elas *mobile* ou não, tornando assim o aplicativo híbrido para ser disposto em qualquer dispositivo, e juntamente a esses *frameworks* foi empregado o banco de dados MySQL para gerenciar as informações. Em suma, o presente trabalho visa facilitar e automatizar o estacionamento da instituição, com maior segurança e tecnologia.

**Palavras Chave:** Aplicativo Híbrido, Automatização, Placas Embarcadas, Comunicação, Internet das Coisas.

**ABSTRACT**

*Parking management can become complex, especially when parking belongs to a plce that develops access to everyone, such as college. This difficulty refres to the validation of access to the same as registration of vehicles, employess and reports that are generated from this information. Weighing in this direction it is necessary to propose na application with a printed communication with na embedded board for na administration of all the data and automation of the control of the parking gates. The required software, all the registrations and validation of inputy and output automatically, but also is able to allow na administrator of all processes manually if there are power outages or failles with the equipment deployed for the operation of the gate. I used in front-end development framework Ionic 2 which is a framework that uses componentes created in HTML5 and CSS3 already shaped for the mobile scenario, in turn without back-end framework AngularJS, a framework maintained by Google that I can be compiled on several platforms, are they mobile or not, thus becoming the hybrid appliaction to be laid out ona ny device, and together with frameworks or employees or MySQL database to manage as information, In summary, the presente work aims to facilitate and automate the parking of the company, with greater security and technology.*

**Keywords:** *Hybrid Application, Automation, Embedded Cards, Communication, Internet of Things.*

**LISTA DE ILUSTRAÇÕES**

Figura 1: Modelagem da base de dados 9

Figura 2: Diagramação Conceitual da entrada de um automóvel 11

Figura 3 - Diagramação Conceitual da saída de um automóvel 12

Figura 4 - Diagramação Conceitual da entrada de um automóvel 14

Figura 5 - Tela de login do sistema 19

Figura 6 - Tela do menu principal 19

Figura 7 - Cadastro de Usuário 20

Figura 8 - Cadastro de Veículo 20

Figura 9 - Procura de Usuário 21

Figura 10 - Alteração de Motorista 21

Figura 11 - Relatório de entrada e saída 22

Figura 12 - Liberação Manual da Portaria 22

Figura 13 - Placa Arduino Uno 23

Figura 14 - Circuito da placa 23

**SUMÁRIO**

[1. INTRODUÇÃO 5](#_Toc498957110)

[2. OBJETIVOS 6](#_Toc498957111)

[2.1 Geral 6](#_Toc498957112)

[2.2 Objetivos Específicos 6](#_Toc498957113)

[3. DESENVOLVIMENTO 8](#_Toc498957114)

[3.1. Requisitos Principais 8](#_Toc498957115)

[3.2. Especificação 9](#_Toc498957116)

[3.3. Comunicação 9](#_Toc498957117)

[4. IMPLEMENTAÇÃO 10](#_Toc498957118)

[4.1. Técnicas e Ferramentas utilizadas 10](#_Toc498957119)

[4.2. Implantação 11](#_Toc498957120)

[4.3.1 Dicionário de dados 13](#_Toc498957121)

[4.4 Diagrama de Caso de Uso 15](#_Toc498957122)

[4.4.1 Descrição dos casos de uso 16](#_Toc498957123)

[4.5 Telas do Sistema 19](#_Toc498957124)

[4.5.1 Tela de Login ao sistema 19](#_Toc498957125)

[4.5.2 Menu Principal 20](#_Toc498957127)

[4.5.3 Cadastro de Usuário 20](#_Toc498957128)

[4.5.4 Cadastro de Carro 21](#_Toc498957129)

[4.5.5 Procurar Usuário 21](#_Toc498957130)

[4.5.6 Alteração de Motorista 22](#_Toc498957131)

[4.5.7 Relatório de movimentação 23](#_Toc498957132)

[4.5.8 Liberação de Portaria Manual 23](#_Toc498957133)

[5. PROGRAMAÇÃO DA PLACA EMBARCADA 23](#_Toc498957134)

[5.1 Placa Embarcada 23](#_Toc498957135)

[**5.2 Circuito 24**](#_Toc498957136)

[**5.3 Funcionamento 25**](#_Toc498957137)

[6. CONCLUSÃO 25](#_Toc498957138)

[REFERÊNCIAS 26](#_Toc498957139)

# 1. INTRODUÇÃO

A Internet das Coisas tem deixado de ser um paradigma não só na área da tecnologia, mas em todas que envolvem o compartilhamento de informações e rastreio de dados, ela tornou-se uma realidade e cada vez mais está tomando conta de dispositivos simples que muitas vezes ficam despercebidos. Dispositivos com poder computacional que possuem essa característica de serem invisíveis podem ser definidos como fazendo parte do conceito de computação ubíqua (GREENFIELD, 2006).

Computação ubíqua foi vislumbrada há mais de uma década atrás. Nesse pensamento computadores habitariam os mais triviais objetos: xícaras de café, canetas, etiquetas de roupa, interruptores de luz, de forma tão invisível para o usuário. Segundo esse conceito seria necessário aprender como conviver com dispositivos computacionais invisíveis e não só como efetuar interações (WEISER, 2017).

O termo de computação Ubíqua diz que qualquer dispositivo que esteja conectado com um alto nível de embarcação ao ponto de ser transparente para a percepção do ser humano é o ápice da inserção da tecnologia no nosso cotidiano, esta ideia tem se amadurecido tanto que acabou gerando as *SmartCities*. (DOMINGUES, FABIANO. 2008).

A definição de *SmartCities* diz que é necessária uma base para um crescimento inteligente e de forma planejada. Utilizar tecnologias para transformar uma comunidade focando em TI para evoluir a qualidade de vida (PREFEITURA DE SÃO PAULO, 2017).

Desta forma a curva de aprendizado torna-se muito maior devido a englobamento de outros fatores importantes, como a gestão de todas essas informações, transmissão, captação e principalmente o armazenamento. Ilustrando a complexidade dessa tarefa e a quantidade de dados que uma “coisa” pode possuir observe uma porta, esse objeto possui propriedades como cor, material, marca, localização e modelo, podendo também possuir dados de status como aberta ou fechada, vida útil, tempo em que ficou aberta. Com Internet das Coisas é possível ter acesso a toda essa massa de dados de qualquer localidade e não se limitar apenas a postas, mas o gerenciamento de toda uma residência e até o controle de uma cidade por inteira.

## 2. OBJETIVOS

## 2.1 Geral

O objetivo deste trabalho é adquirir conhecimento científico e desenvolver um software para automação do estacionamento da FATEC Garça por meio de sistemas embarcados e híbridos, com o auxílio da concepção sobre Internet das Coisas.

## 2.2 Objetivos Específicos

* Desenvolver um módulo de cadastro de carros que tem acesso à faculdade e realizar a validação e entrada ou saída dos mesmos automaticamente.
* Desenvolver um módulo de cadastro de usuários em diferentes hierarquias para controle de acessos a faculdade (Aluno, Professor, Administrador e outros);
* Desenvolver uma biblioteca que seja apta capturar imagens (placas de automóveis) e transformá-las em algarismos para consultas no sistema;
* Efetuar o vínculo das informações geradas pelo sistema (horário de entrada e saída, carro e usuário);
* Desenvolver um *web-service* capaz efetuar comunicação entre o sistema híbrido de controle e o circuito embarcado;
* Desenvolver um circuito que seja capaz de realizar o controle das cancelas do estacionamento quando o mesmo for acionado pelo *Web Service*;
* Um *software* que esteja disponível tanto localmente (*desktop*), quanto hospedado na nuvem ou *mobile*;
* Geração de relatórios de entrada e saída;
* Registros de log para análise do administrador.

# 3. DESENVOLVIMENTO

Neste capítulo serão apresentados os principais aspectos relacionados ao desenvolvimento.

## 3.1. Requisitos Principais

Para o desenvolvimento do aplicativo para gerenciamento de um estacionamento foram identificados os seguintes requisitos:

* Cadastro completo de Usuários que tem acesso ao estacionamento, dividido por categorias (Aluno, Professor e Outros);
* Cadastro completo de Carros que estejam vinculados aos usuários;
* Gerar relatórios de entrada e saída do estacionamento com data, horário e outras informações pertinentes;
* Validar informações automaticamente com a aproximação de um veículo e registra-la no log;
* Captura e conversão dos dados pela câmera instalada na cancela do estacionamento;
* Execução do circuito eletrônico quando a validação for efetuada;

O *software* conta com o manuseio de todas as funcionalidades manualmente caso ocorra algum tipo de interrupção seja por falha técnica ou razões que não habilitem os componentes eletrônicos para validação e inserção das informações de entradas e saídas.

Existe apenas um tipo de usuário que tem acesso ao sistema, este é o administrador, as outras entidades se baseiam em um cadastro padrão que difere de seus tipos como alunos, professores e outros, para constarem em relatórios e vínculos que podem ser feitos dentro do sistema.

## 3.2. Especificação

Para especificação deste projeto utilizou-se a *Unified Modeling Language (UML)*. Foi utilizado a ferramenta *MySQL Workbench* para modelagem de dados e *Astah Community* para a modelagem do software.

O software foi desenvolvido usando o *framework* Ionic 2 e AngularJS. Corresponde a um ambiente de desenvolvimento de alta performance, orientado a objeto, com arquitetura de software MVC. Além disso, o Ionic e o AngularJS contam com inúmeras funcionalidades de segurança e facilidades para compilação em diferentes hardwares.

Para armazenamento das informações foi escolhido o banco de dados relacional MySQL sendo um dos bancos mais utilizados e bastante aceito em servidores de hospedagem facilitando sua implementação.

## 3.3. Comunicação

Um dos grandes obstáculos para o desenvolvimento deste projeto foi interligar todas as plataformas para que as mesmas obtivessem uma comunicação satisfatória sem percas de sinais e agilidade no processo.

No projeto atual foi utilizado o protocolo de comunicação *Messege Queue Telemetry Transporty* (MQTT) desenvolvido e mantido pela IBM com um cabeçalho fixo de 2 bytes e até 12 bytes de informações variáveis adicionais, um protocolo ágil, de baixa armazenagem e ideal para utilizar com M2M (*Machine-To-Machine*).

Trata-se de um protocolo de mensagens baseado na arquitetura *publish/subscribe*, voltado para dispositivos restritos e redes inseguras, com baixa largura de banda e alta latência (STEVENTON, 2006).

Modelo de entidade de relacionamento é modelo mais simples, mas com uma visão dos relacionamentos entre os objetos, baseado na percepção de mundo real, que consiste em uma coleção de objetos básicos, chamados entidades, e em relacionamentos (GIUSTO, 2010).

Na Figura 1 é mostrado a modelagem do banco de dados do sistema, a mesma base de dados será usada para *mobile*, *desktop* e *site*.

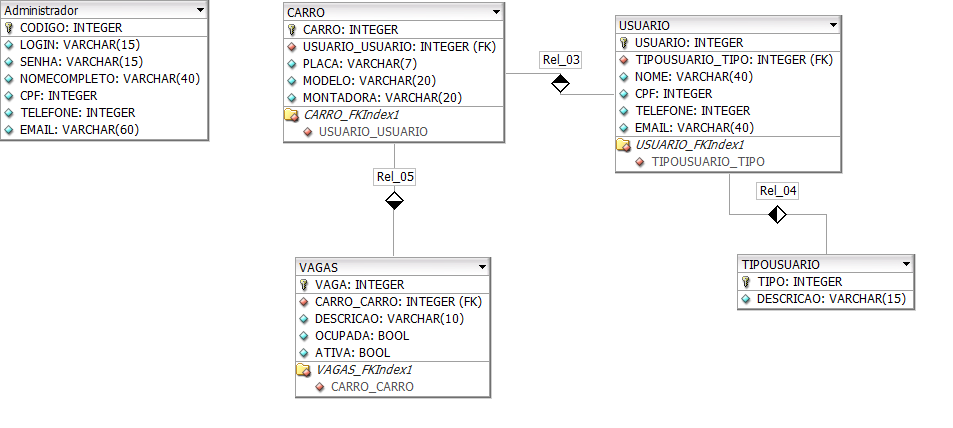


Figura 1: Modelagem da base de dados

É possível notar simples relacionamentos entre tabelas, o sistema deve realmente ter o conceito de simplicidade, não é algo que necessite esforço do usuário que estiver responsável pelos cadastros e autenticações, onde todo o processo será feito no momento das validações no *WebService* enviado pela câmera posicionada na cancela do estacionamento.

Para que todo o processo ocorra automaticamente se faz imprescindível a eficácia da comunicação do sistema com a placa controladora e a clareza nas imagens capturadas pela câmera e transformadas em dados para posterior análise.

## 4. IMPLEMENTAÇÃO

Neste capítulo será mostrando as formas que será o desenvolvimento do Aplicativo Híbrido Para Gerenciamento De Um Estacionamento.

## 4.1. Técnicas e Ferramentas utilizadas

Para a criação da interface será utilizada as linguagens de HTML5, CSS e JavaScript que estão englobadas no *framework* Ionic 2. A comunicação com o banco de dados será realizada através da linguagem PHP.

## 4.2. Implantação

A implantação do software terá que ser feita em um servidor PHP com banco de dados MYSQL. O servidor terá que atender as demandas de envio de arquivos, processamento de informações e retorno de funções rapidamente, por isso a velocidade e qualidade da implantação devem ser levadas em consideração.

**4.3. Diagramação de atividade da aplicação**

No contexto da UML, o Diagrama de Atividades é um diagrama comportamental (que especifica o comportamento do *software*), e através dele podemos modelar partes do comportamento de um *Software*.

O diagrama de atividades ilustra graficamente como será o funcionamento do *software* (em nível micro ou macro), como será a execução de alguma de suas partes, como será a atuação do sistema na realidade de negócio na qual ele está inserido.

Na Figura 2, tem-se a diagramação conceitual das atividades da entrada de um veículo no Aplicativo Híbrido Para Gerenciamento De Um Estacionamento.

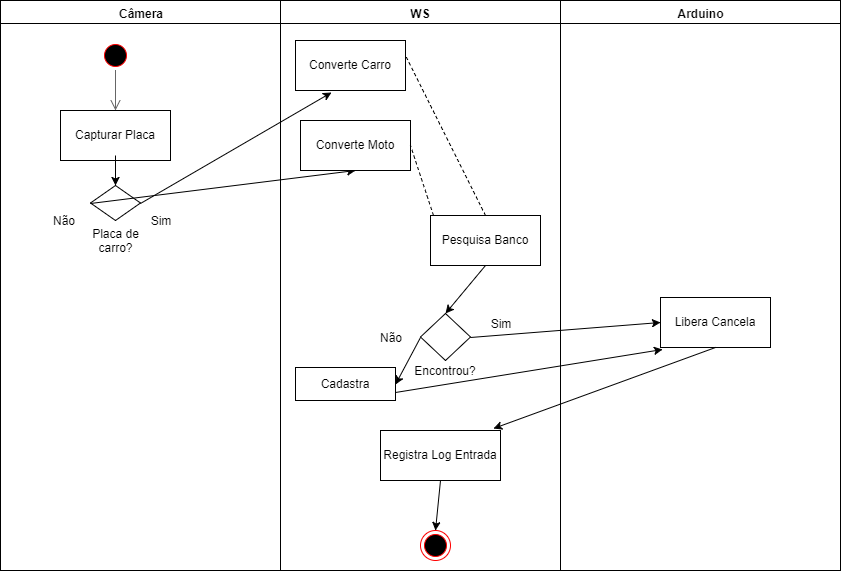
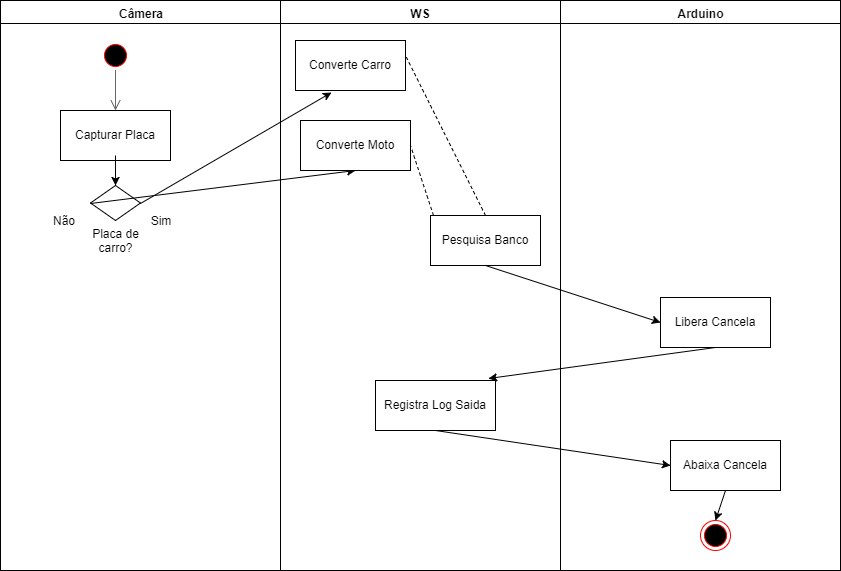
Na Figura 3, tem-se a diagramação conceitual das atividades da saída de um veículo no Aplicativo Híbrido Para Gerenciamento De Um Estacionamento.

Figura 2: Diagramação Conceitual da entrada de um automóvel

Figura 3 - Diagramação Conceitual da saída de um automóvel

## 4.3.1 Dicionário de dados

Este é um documento onde um analista tem as informações de todos os objetos do modelo e sua estrutura de uma forma contextual é neste documento que obtém detalhes que o diagrama de entidades não deixa claro.

Neste tópico serão mostrados os dicionários de dados das entidades relacionadas do software.

**4.3.1.1 Entidade admin**

Entidade armazena os dados dos administradores que serão responsáveis pela inserção das informações, extração de relatórios e se necessário, validações de entradas e saídas do estacionamento.

Tabela 1 : Entidade admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Campo** | **Descrição** | **Tipo** | **NN** |
| PK | codigo | Identificação do administrador | INT(11) | \* |
|  | login | Login do administrador | VARCHAR(15) | \* |
|  | senha | Senha do administrador | VARCHAR(15) | \* |
|  | nomecompleto | Nome do usuário logado | VARCHAR(40) |  |
|  | cpf | CPF do usuário logado | VARCHAR(11) | \* |
|  | telefone | Telefone do usuário logado | VARCHAR(11) |  |
|  | email | E-mail do usuário logado | VARCHAR(60) |  |

**4.3.1.2 Entidade carro**

Entidade armazena os dados dos carros que serão cadastrados pelos administradores do sistema, o principal objetivo com esta entidade é conseguir realizar consultas para validações do acionamento da cancela.

Tabela 2 : Entidade carro

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Campo** | **Descrição** | **Tipo** | **NN** |
| PK | carro | Código de controle do carro | INT(11) | \* |
| FK | usuario | Código do usuário que está vinculado ao automóvel | INT(11) | \* |
|  | placa | Placa do automóvel | VARCHAR(7) | \* |
|  | modelo | Modelo do automóvel | VARCHAR(20) |  |
|  | montadora | Montadora do automóvel | VARCHAR(20) |  |

**4.3.1.3 Entidade usuario**

Entidade armazena os dados dos usuários do estacionamento, essencial para geração do relatório dos tipos de usuário que mais frequentam a faculdade.

Tabela 3 : Entidade usuario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Campo** | **Descrição** | **Tipo** | **NN** |
| PK | usuario | Código do usuário | INT(11) | \* |
| FK | tipousuario | Código do tipo do usuário | INT(11) | \* |
|  | nome | Nome do usuário | VARCHAR(40) | \* |
|  | cpf | Cpf do usuário | VARCHAR(11) | \* |
|  | telefone | Telefone do usuário | VARCHAR(11) |  |
|  | email | Email do usuário | VARCHAR(60) |  |

**4.3.1.4 Entidade tipousuario**

Entidade armazena os tipos de usuário para vínculo no cadastros dos mesmos.

Tabela 4 : Entidade tipousuario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Campo** | **Descrição** | **Tipo** | **NN** |
| PK | tipo | Código do tipo do usuário | INT(11) | \* |
|  | descricao | Descrição do tipo do usuário | VARCHAR(15) | \* |

**4.3.1.5 Entidade relentsai**

Entidade armazena o fluxo de entrada e saída do estacionamento.

Tabela 5 : Entidade relentsai

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Campo** | **Descrição** | **Tipo** | **NN** |
| PK | codigo | Código incremental de controle | INT(11) | \* |
| FK | usuario | Código do usuário que está acessando o estacionamento | INT(11) | \* |
|  | datahora | Data e hora da ocorrência | DATETIME | \* |
|  | tipo | Tipo da ocorrência (entrada ou saída) | CHAR(1) | \* |

## 4.4 Diagrama de Caso de Uso

Diagrama de caso de uso fornece em modo de descrever a visão externa do sistema e suas interações do mundo exterior, e o seu propósito é descrever os requisitos funcionais do sistema e uma descrição clara sobre as responsabilidades que devem ser cumpridas pelo sistema (FURLAN,1998).

Na Figura 4, O diagrama de casos de uso representa as ações do administrador, e logo a seguir algumas descrições de caso de uso mais relevantes do sistema.

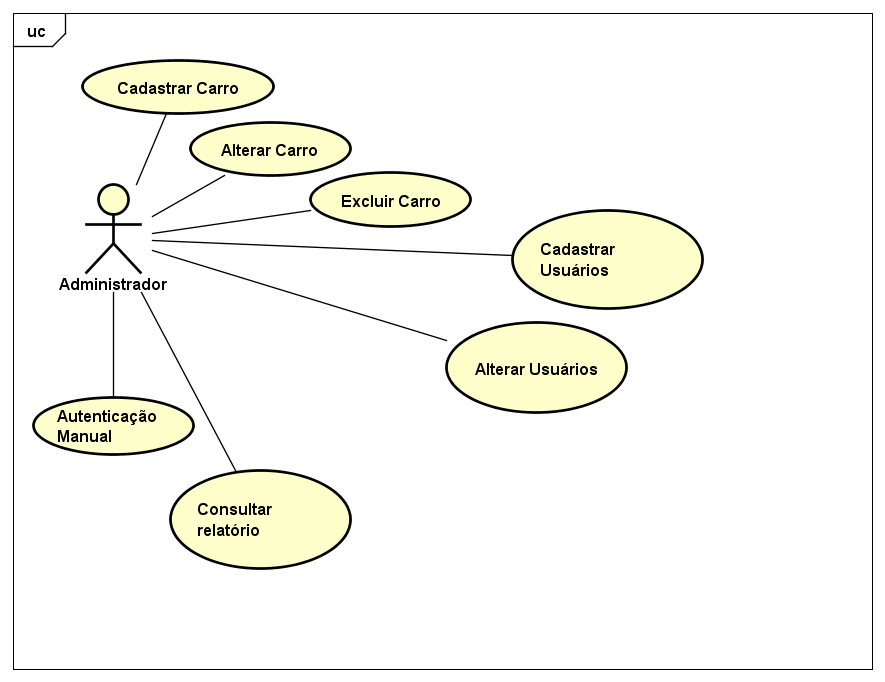


Figura 4 - Diagramação Conceitual da entrada de um automóvel

## 4.4.1 Descrição dos casos de uso

**Caso de uso:** Cadastrar Usuário

**Descrição:** O Administrador deverá informar os dados cadastrais dos usuários (motoristas) que irão ter acesso ao estacionamento.

**Atores envolvidos:** Administradores e Usuários.

**Pré-condição:** Não existe.

**Cenário Principal de Sucesso:**

1. O caso de uso é iniciado com o administrador informando os dados cadastrais do usuário.

2. O administrador deverá selecionar o tipo do usuário que está sendo inserido.

* 1. O banco de dados irá realizar uma pesquisa sobre os tipos de usuários.

3. O administrador clica no botão gravar.

1. O banco de dados verifica se o CPF informado já não existe na base.

4. O usuário é cadastrado.

**Cenário Secundário de Falha:**

2.1.1. Não será possível realizar o cadastro, pois serão necessários tipos de usuários informados.

3.1.1 O sistema retorna a mensagem de CPF já cadastrado.

**Caso de uso:** Alterar Usuário

**Descrição:** O Administrador deverá informar os dados cadastrais dos usuários (motoristas) para alteração dos mesmos.

**Atores envolvidos:** Administradores e Usuários.

**Pré-condição:** Usuário já ter cadastro.

**Cenário Principal de Sucesso:**

1. O caso de uso é iniciado com o administrador informando os dados cadastrais do usuário.

2. O administrador deverá selecionar o tipo do usuário que está sendo alterado ou manter o mesmo.

* 1. O banco de dados irá realizar uma pesquisa sobre os tipos de usuários.

3. O administrador clica no botão gravar.

1. O banco de dados verifica se o CPF informado já não existe na base.

4. O usuário é alterado.

**Cenário Secundário de Falha:**

2.1.1. Não será possível realizar a alteração, pois serão necessários tipos de usuários informados.

3.1.1 O sistema retorna a mensagem de CPF já cadastrado.

**Caso de uso:** Cadastrar Carro

**Descrição:** O Administrador deverá informar os dados cadastrais dos carros que irão ter acesso ao estacionamento.

**Atores envolvidos:** Administradores.

**Pré-condição:** Usuário cadastrado para vínculo com veículo.

**Cenário Principal de Sucesso:**

1. O caso de uso é iniciado com o administrador informando os dados cadastrais do veículo.

2. O administrador deverá selecionar qual usuário está vinculado com aquele veículo.

* 1. O banco de dados irá realizar uma pesquisa sobre os usuários cadastrados.

3. O administrador clica no botão gravar.

4. O veículo é cadastrado.

**Cenário Secundário de Falha:**

2.1.1. Não será possível realizar o cadastro, pois é necessário informar o usuário vinculado.

**Caso de uso:** Alterar Carro

**Descrição:** O Administrador deverá informar os dados cadastrais dos veículos para alteração dos mesmos.

**Atores envolvidos:** Administradores.

**Pré-condição:** Usuário cadastrado para vínculo com veículo e veículo já cadastrado.

**Cenário Principal de Sucesso:**

1. O caso de uso é iniciado com o administrador informando os dados cadastrais do veículo.

2. O administrador deverá selecionar qual usuário está vinculado com aquele veículo.

* 1. O banco de dados irá realizar uma pesquisa sobre os usuários cadastrados.

3. O administrador clica no botão gravar.

4. O veículo é alterado.

**Cenário Secundário de Falha:**

2.1.1. Não será possível realizar o cadastro, pois é necessário informar o usuário vinculado.

**Caso de uso:** Excluir Carro

**Descrição:** Neste caso de uso o Administrador acessa a pesquisa de carros, escolhe o veículo e pode excluir o mesmo.

**Atores envolvidos:** Administradores.

**Pré-condição:** Carro cadastrado.

**Cenário Principal de Sucesso:**

1. O caso de uso é iniciado com o administrador acessando a tela de veículos.

2. O administrador deverá selecionar o veículo que deseja que seja deletado.

3. O administrador clica no botão excluir.

4. O veículo é excluído.

**Cenário Secundário de Falha:**

Não existe.

**Caso de uso:** Consultar Relatório

**Descrição:** Neste caso de uso o Administrador acessa a página de emissão de relatórios, filtra períodos e avalia o mesmo.

**Atores envolvidos:** Administradores.

**Pré-condição:** Não existe.

**Cenário Principal de Sucesso:**

1. O caso de uso é iniciado com o administrador acessando a tela de relatórios.

2. O administrador deverá selecionar um período para filtrar.

3. O administrador clica no botão pesquisar.

3.1 O sistema busca no banco de dados por todo o fluxo do estacionamento no período selecionado.

4. O relatório é emitido.

**Cenário Secundário de Falha:**

3.1.1 Caso não exista fluxo o sistema retorna a mensagem de fluxo inexistente.

**Caso de uso:** Autenticação Manual

**Descrição:** Neste caso de uso o Administrador acessará a página de autenticação para liberar o acesso de carros manualmente.

**Atores envolvidos:** Administradores.

**Pré-condição:** Carro cadastrado.

**Cenário Principal de Sucesso:**

1. O caso de uso é iniciado com o administrador acessando a tela de autenticação manual.

2. O administrador deverá selecionar o veículo que será validado.

3. O administrador clica no botão autenticar.

4. O veículo é autenticado.

**Cenário Secundário de Falha:**

Não existe.

## 4.5 Telas do Sistema

Neste capítulo será mostrando algumas telas principais de interação do Aplicativo para Gerenciamento de um Estacionamento.

## 4.5.1 Tela de Login ao sistema

Nesta tela o administrador informa seu usuário e senha se o mesmo for valido ele tem acesso ao sistema, senão é exibida uma mensagem de erro de validação.

## 

Figura 5 - Tela de login do sistema

## 4.5.2 Menu Principal

Nesta tela o administrador poderá escolher entre cadastrar usuários, vincular carros, procurar usuário, liberar a portaria manualmente ou automaticamente, relatório de movimentação e trocar de usuário.

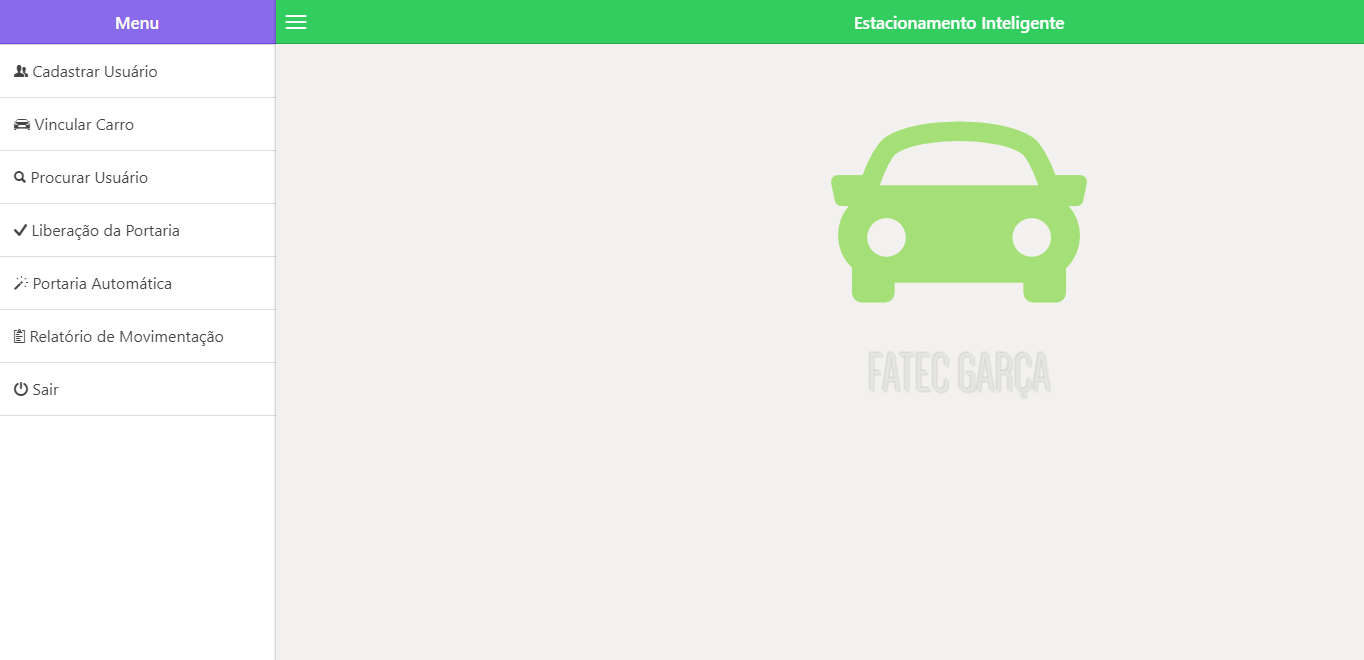


Figura 6 - Tela do menu principal

## 4.5.3 Cadastro de Usuário

Nesta tela o administrador cadastra um usuário que terá acesso ao estacionamento para posteriormente ser vinculado a um veículo.

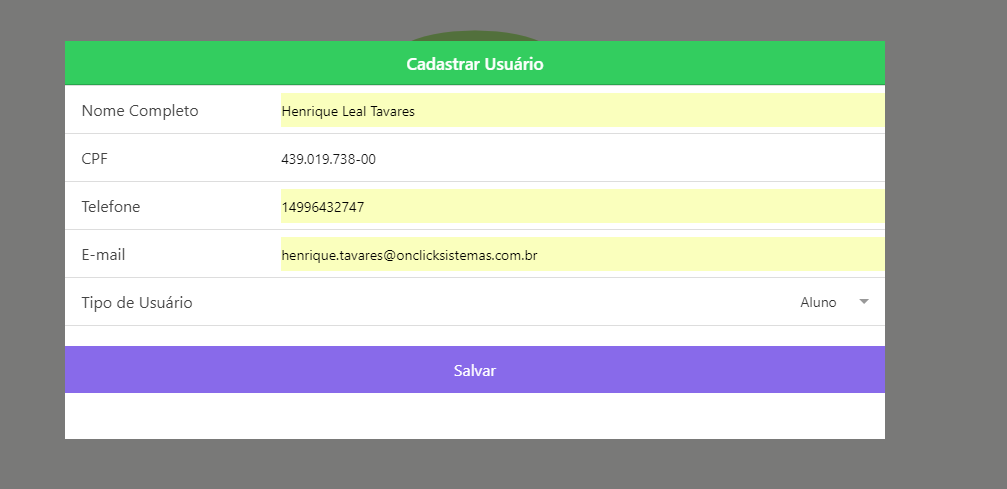


Figura 7 - Cadastro de Usuário

## 4.5.4 Cadastro de Carro

Nesta tela o administrador vincula um veículo a um usuário.

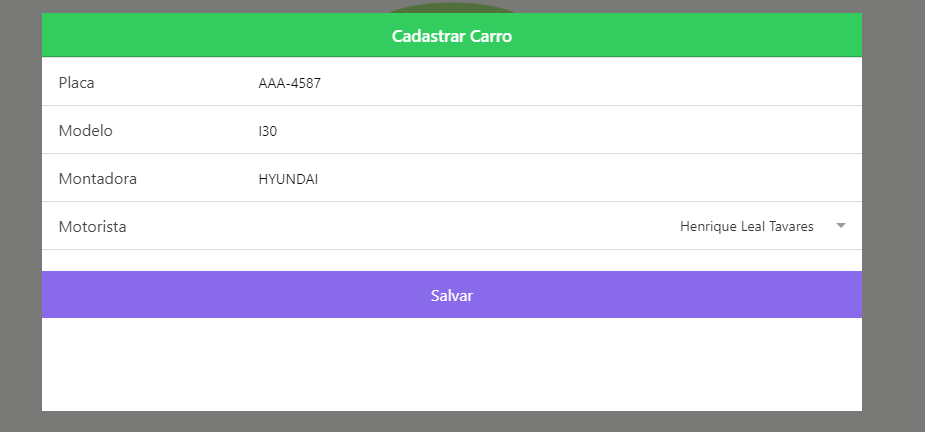


Figura 8 - Cadastro de Veículo

## 4.5.5 Procurar Usuário

Nesta tela o administrador consegue realizar filtros por modelo, placa, tipo de usuário e nome do mesmo para possíveis alterações ou consultas



Figura 9 - Procura de Usuário

.

## 4.5.6 Alteração de Motorista

Nesta tela o administrador pode fazer a alteração dos dados do motorista, se o mesmo for excluído seus veículos vinculados também serão deletados.

Figura 10 - Alteração de Motorista

## 4.5.7 Relatório de movimentação

Nesta tela o administrador poderá realizar filtros e verificar o fluxo de entradas e saídas de veículos em um determinado período.

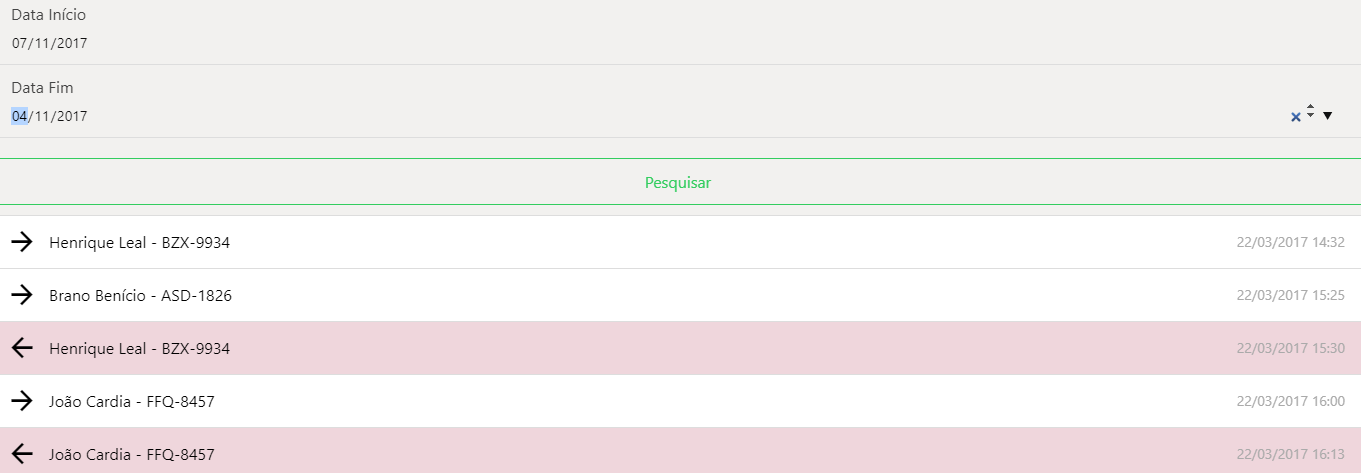


Figura 11 - Relatório de entrada e saída

## 4.5.8 Liberação de Portaria Manual

Nesta tela será possível o administrador identificar o usuário que está tentando acessar o estacionamento e registrar o mesmo manualmente.

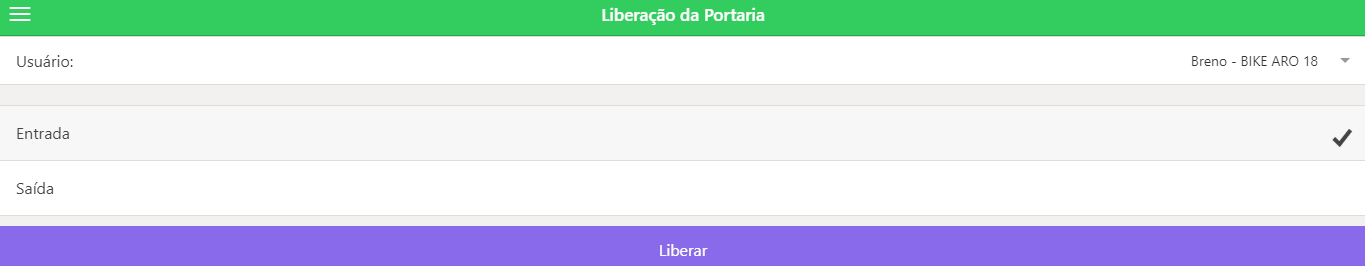


Figura 12 - Liberação Manual da Portaria

## 5. PROGRAMAÇÃO DA PLACA EMBARCADA

## 5.1 Placa Embarcada

Para realização deste projeto foi escolhida a placa Arduino Uno, que servirá de apoio na automação da cancela do estacionamento.

O Arduino teve o início de seu desenvolvimento em Ivrea, Italia em 2005, com o intuito de criar ambientes de interação em projetos acadêmicos como uma alternativa de um menor orçamento comparado aos outros tipos de prototipagens disponíveis na época (Arduino.com).

Recebeu uma menção honrosa na categoria *Digital Communitys* em 2006, pela Prix Ars Eletronic, além de atingir à marca de mais de cinquenta mil placas vendidas até outubro de 2008, onde o dispositivo ainda estava sendo desenvolvido.

A Figura 13 mostra a versão da placa Arduino que foi utilizada neste projeto:

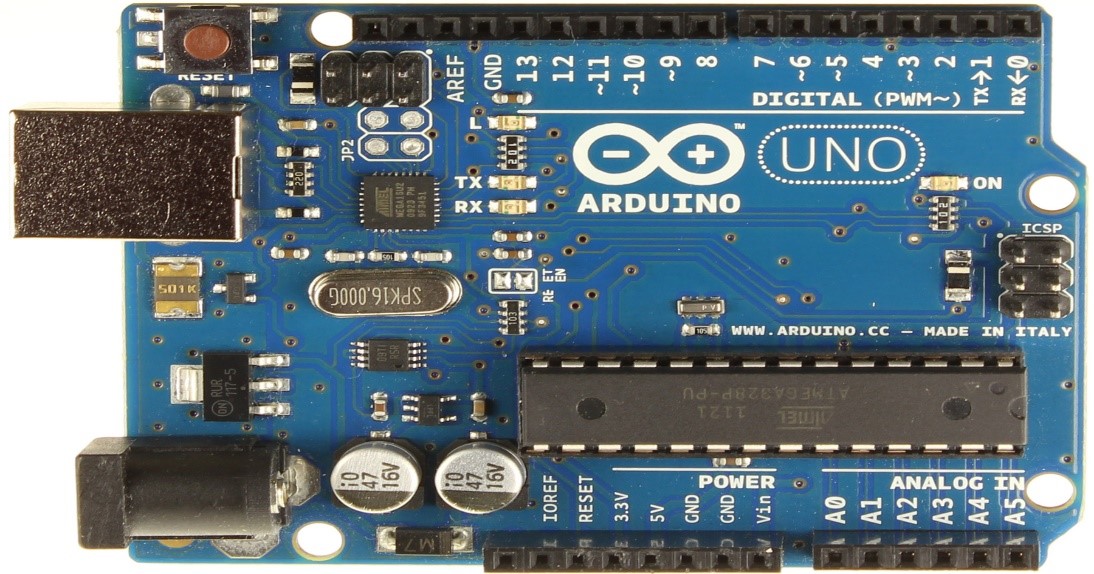


Figura - Placa Arduino Uno

### 5.2 Circuito

A figura 14 mostra o circuito que será utilizado para contemplar o desenvolvimento.

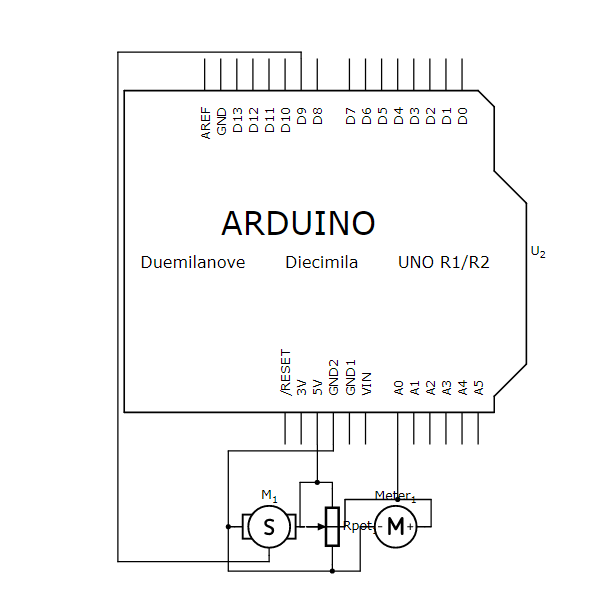


Figura - Circuito da placa

Foram utilizados diversos componentes eletrônicos, todos eles interligados em uma *protoboard* simples, entre eles:

* 1 Arduino Uno
* 7 Jumpers macho/fêmea
* 1 Led RGB
* 1 Shield Ethernet - PCS ENC28J60
* 1 Micro Servo 9g SG90
* 1 Protoboard Simples

### 5.3 Funcionamento

A comunicação entre o sistema e a placa será somente através do Shield Ethernet, o mesmo é capaz de hospedar uma página e gerar um protocolo IP para acesso externo, assim que a placa do automóvel for validada o sistema irá acessar este IP e enviar um comando via *Web-Service* para a liberação da cancela.

## 6. CONCLUSÃO

Com o Aplicativo para Gerenciamento de um Estacionamento é possível a instituição controlar todo seu fluxo de acesso ao seu estacionamento, verificando periodicamente a sazonalidade e os tipos de usuários que mais o acessam, aumentando assim a segurança, pois para acessar a unidade é necessário ter um cadastro tanto de dados pessoais quanto do veículo previamente registrado, concomitantemente a automação da cancela, que é o principal objetivo deste projeto, facilita a validação de entrada e saída do mesmo.

# REFERÊNCIAS

GREENFIELD, Adam. Everyware The Dawning age of ubiquitous computing. [S.I.]: New Riders, 2006.

WEISER, Mark. Ubiquitous Computing. Disponível em:

< http://www.ubiq.com/hypertext/weiser/UbiHome.html>. Acesso em: 17 de novembro de 2017.<http://www.ubiq.com/hypertext/weiser/UbiHome.html>. Acesso: 09 de outubro de 2017.

DOMINGUES, Fabiano. Computação Ubíqua, página online disponível em: <http://www.hardware.com.br/artigos/computacao-ubiqua>. Acesso em: 6 de outubro de 2017.

PREFEITURA DE SÃO PAULO. Secretaria de Serviços trabalha para tornar São Paulo uma cidade inteligente. Disponível em: <http://www.prefeitura.sp.gov.br/cidade/secretarias/inovacao/noticias/?p=177639>. Acesso em: 09 de outubro de 2017.

A. Steventon and S. Wright, Intelligent Spaces*:* The Application of Pervasive ICT. Computer Communications and Networks. Springer Londres, 2006. Disponível em: <https://books.google.com. br/books?id=C7MGmQEACAAJ>. Acesso em : 17 de Novembro de 2017.

D. Giusto, A. Iera, G. Morabito, and L. Atzori, The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications. Springer New York, 2010. Disponível em: <https://books.google.com.br/books?id= vUpiSRc0b7AC>. Acesso em: 17 de Novembro de 2017.