CENTRO PAULA SOUZA

FATEC OURINHOS

CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

FELIPE CORRÊA GOMES

GABRIEL DE MORAES TORRES

**SISTEMA PARA CONTROLE DE ESTACIONAMENTOS**

OURINHOS-SP

2019

FELIPE CORRÊA GOMES

GABRIEL DE MORAES TORRES

**SISTEMA PARA CONTROLE DE ESTACIONAMENTOS**

Trabalho de Graduação apresentada à Faculdade de Tecnologia de Ourinhos para conclusão do curso de Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Wiliam Carlos Galvão

OURINHOS-SP

2019

Folha de aprovação

Dedicatória P

Dedicatória S

Agradecimentos

RESUMO

Esse trabalho tem o objetivo de apresentar o desenvolvimento do sistema para o controle da disponibilidade de vagas do estacionamento via wireless pela necessidade de agilizar o fluxo de veículos e diminuir os problemas encontrados. Utilizando o Arduino e os respectivos módulos para o procedimento de conexão com a rede e de verificação de proximidade com infravermelho. O desenvolvimento do sistema utilizará a linguagem de programação C++ para o controle do Arduino e o processamento de dados web PHP.

Palavras-chave: Arduino, estacionamento, vagas.

ABSTRACT

This paper has the objective to present the development of system to control the availability of parking lots, wireless by the necessity to speed up the flow of vehicles and reduce the problems encountered. Using Arduino and the related modules for the procedure of connection with the web and the verification of proximity with infra-red. The development of the system will use the programming language C++ for the Arduino control and the data processing web PHP.

Key word: Arduino, parking, vacancies.

Lista de Figuras

Figura 1 – Lei de Ohm 18

Figura 2 – Tela Inicial do Arduino IDE 25

Figura 3 – Tela inicial Fritzing 26

Figura 4 – Tela inicial NetBeans 27

Figura 5 - Kit Arduino Básico 28

Figura 6 – Sensor de proximidade de infravermelho 29

Figura 7 – Esquema do circuito do sensor de proximidade 29

Figura 8 – Modelo Cascata 30

Figura 9 – Modelo de funcionamento 32

Figura 10 – Modelo de funcionamento do circuito 32

Figura 11 - Cronograma 34

Figura 12 – Tela de *Login* 46

Figura 13 – Tela de vagas para *Desktop* 46

Figura 14 – Tela de vagas para Celulares 47

Sumário

1Introdução 12

2Revisão Bibliográfica 14

2.1Microcontroladores 14

2.2Infravermelho 14

2.3Arduino 15

2.3.1 Arduino pro micro Leonardo 16

2.3.2 Módulo Receptor Tsop1832 16

2.3.3 Módulo Emissor TIL32 16

2.3.4 Módulo ESP8266 17

2.3.5 NodeMCU 17

2.4Lei de Ohm 18

2.5Rede 18

*2.5.1* *Wireless*  19

2.5.2 Protocolos 19

2.6Sistema *web*  20

2.6.1 HTML 20

2.6.2 CSS 21

2.6.3 PHP 21

2.6.4 MySql 21

2.7C++ 22

2.8Trabalhos correlatos 22

2.8.1 Sistema online para estacionamento de veículos 22

2.8.2 Controle centralizado de equipamentos de ar-condicionado 22

3Metodologia 24

3.1Ferramentas 24

3.1.1 Arduino IDE 24

3.1.2 Fritzing 25

3.1.3 NetBeans 26

3.1.4 Kit Básico do Arduino 27

3.1.5 Sensor de proximidade 28

3.2Procedimentos 29

3.3 Desenvolvimento 30

3.3.1Análise e Definição de Requisitos 31

3.3.2Projeto de Sistema e de *Software*  31

3.3.3Implementação e Teste de Unidades 33

3.3.4Integração e Testes de Sistemas 33

3.3.5Manutenção 33

4Cronograma 34

APÊNDICE A – Documento de Requisitos 37

APÊNDICE B – Diagrama de *Use Case*  42

RF- 02 –Efetuar login 42

RF- 03 –Reservar Vaga 43

RF- 04 – Gerenciar Vaga 43

RF- 05 – Relatar Problema 43

RF- 06 – Gerar Relatório 44

APENDICE D – Protótipos o sistema 46

APENDICE D – Modelo Canvas 48

APENDICE E – Resultados finais 49

1. Introdução

Atualmente alguns dos ambientes que possuem um grande tráfego como em *shoppings*, universidades, ambientes de lazer, entre outros, há necessidade de controlar as vagas disponíveis para que haja maior mobilidade dos condutores.

Com a finalidade de auxiliar no controle das vagas desses ambientes para exibir os locais livres, reservados ou ocupados e agilizar o fluxo de veículos, verifica-se a carência no desenvolvimento de um sistema para gerenciar o estacionamento.

O intuito é utilizar o Arduino para o controle da entrada de dados dos sensores de infravermelho que verificam se há algum veículo próximo e exibir disponibilidade da vaga.

Problema

O atual problema é que na ausência do controle de vagas, não é possível verificar se há lugares livres em um estacionamento, sendo a única alternativa o condutor verificar cada vaga para confirmar que exista alguma disponível. Resultando na insatisfação do condutor para retornar em determinado local ou na perda de clientes indiretos em empresas que não haja esse controle.

Objetivo

O objetivo desse trabalho será desenvolver um sistema para controle de estacionamento em que seja possível verificar o estado da vaga em um determinado local e organizar o ambiente onde é aplicado utilizando antecipadamente um modelo da planta baixa.

Justificativa

Em estacionamentos com um grande fluxo de veículos, foi observado que não há um controle devido das vagas ou esse controle é muito limitado, em que o condutor do veículo é obrigado a procurar uma determinada vaga sem auxilio com incerteza de que haja lugares livres. Necessitando de algum auxílio para diminuir o tempo de espera do condutor, juntamente agilizando o fluxo desses veículos e exibindo os lugares disponíveis.

De acordo com a observação de Junior (2016), o fluxo de automóveis em áreas públicas é alto, especialmente em finais de semana e em feriados ou datas para eventos de lazer. Após fazer a análise foi verificado que os estacionamentos eram trafegados por veículos de visitantes.

Com essa análise supõe-se que os veículos que permeiam necessitam de um sistema para determinar os lugares que possam ocupar, pois a taxa de veículos em feriados e datas comemorativas aumentam causando lentidão e outro tipos de obstáculos.

1. Revisão Bibliográfica

Esse capítulo será descrito as ferramentas utilizadas para o desenvolvimento do trabalho.

* 1. Microcontroladores

Os microcontroladores são dispositivos simples que contém internamente as memórias ROM e RAM, oscilador de *clock*, entre outros sistemas de entrada e saída. Sistemas baseados nesse dispositivo tem a característica de ter melhor custo benefício e contribui pelo tamanho.

O microcontrolador é um pequeno computador embutido em um chip. Contendo alguns recursos básicos como a memória RAM, um processador e uma memória flash para armazenar os programas, entre outros (MONK, 2017).

O microcontrolador que será utilizado é da família AVR da Atmel, mas há uma variedade de microcontroladores que poderá ser manuseado para desenvolver esse projeto como o PIC, COP, 8051 entre outros. A semelhança entre os microcontroladores é grande possibilitando a migração de um microcontrolador para outro tendo sua arquitetura compatível, pois os conceitos entre eles não mudam muito.

Estando de acordo em que:

“Os microcontroladores, ao contrário de seus irmãos microprocessadores, são dispositivos mais simples com memórias RAM e ROM internas, oscilador interno de clock, I/O interno, entre outros, sendo por isso chamados muitas vezes de computadores em um único chip. Tais características tornam mais simples o projeto de dispositivos inteligentes, pois os MCU’s raramente necessitam de CI’s externos para funcionar, o que contribui para diminuição de custos e tamanho “(PEREIRA, 2012, p. 12).

* 1. Infravermelho

Como outras cores existentes a tonalidade de vermelho é uma das cores mais quentes que existe, mas de acordo com pesquisadores há outra tonalidade que é próxima a coloração avermelhada, mas não é visível a olho nu sendo perceptível pela temperatura. Dependendo da cor a radiação é visível, enquanto o comprimento das ondas do infravermelho corresponde de 1 milímetro a 700 nanômetros dando aspecto de ser invisível a olho nu devido ao comprimento maior que as ondas das cores perceptíveis.

O infravermelho tem reação não iônica, pois não é prejudicial à saúde, são referentes as ondas de rádio, as micro-ondas, entre outros tipos não nocivas à saúde.

A distância curta da onda é relativa a 0,5 µm e a longa aproximadamente a 1mm. A onda curta é visível, sendo emitida por todos os tipos de emissores de lus. Sendo captada por meios mais simples como chapas fotovoltaicas. O infravermelho é limitado a uma área restringida, pois o receptor dever estar em direção ao emissor.

Segundo Comer (2007), a comunicação de controles remotos sem fio como os utilizados em televisores ou os de aparelhos de som comunicam-se através de um sistema de infravermelho.

* 1. Arduino

O Arduino é uma plataforma de prototipagem eletrônica *open-source* e *open-hardware* equipado com microcontroladores AVR Atmel ou microprocessadores Intel. Tem a funcionalidade de um pequeno computador, em que controla as entradas e saídas do *hardware*. Possuindo as mesmas funções do MCU em um circuito integrado havendo o controle do oscilador de *clock*, controle de portas I/O, controle de surtos de tensão, entre diversas características representadas em cada modelo específico.

O Arduino é a melhor escolha para desenvolvimento de projetos de pequeno a grande porte, atualmente ele auxilia em diversas áreas, incluindo na arquitetura de big data, na computação em nuvem, no desenvolvimento de drones, reconhecimento biométrico, etc.

McRoberts (2015) afirma que o Arduino em comparativo com outros microcontroladores é mais fácil para utilizar, pois não é necessário ser da mesma área técnica para desenvolver os próprios projetos, e agiliza o intervalo de tempo do desenvolvimento.

De acordo com Monk (2017) O sucesso de usuários para o desenvolvimento no Arduino é a mobilidade de utilizar outros módulos ou *shields* sendo necessário apenas encaixar ou empilhar esses módulos para agregar determinada funcionalidade ao Arduino.

1. Arduino pro micro Leonardo

Apesar desse modelo do Arduino ter as mesmas funcionalidades básicas que os outros, a portabilidade é maior referente ao consumo de energia e o tamanho. Vem acompanhado do suporte ao USB micro b para facilitar na conexão direta com o computador ou qualquer tipo de fonte de energia que tenha o mesmo conector e contendo o microcontrolador ATmega32U4 com *clock* de processamento de 16 mHz tendo abundância de processamento.

MONK (2017) afirma que, essa placa é mais acessível por conter uma interface USB integrado ao chip sem necessitar de um controlador externo. E contém a mesma característica que outros modelos, devendo ter a capacidade de programação na mesma linguagem para facilitar a migração de um modelo para o outro.

1. Módulo Receptor Tsop1832

O módulo receptor Tsop1832 representa um sistema de circuito integrado específico que recebe as ondas de infravermelho, nele contém um diodo PIN, um pré-amplificador de ondas e uma camada de epóxi servindo como filtro do infravermelho.

Com a frequente utilização desse modulo em diversos tipos de circuitos, ele contém um controle de surtos de tensão e de frequência para não comprometer seu funcionamento. A frequência de funcionamento entre os modelos do módulo está entre os 30 kHz e 56 kHz, referente aos dois últimos dígitos esse que será utilizado é o modelo de 32 kHz, para reagir a mesma frequência que o módulo emissor.

Vishay (2013) afirma que, esse determinado módulo é um receptor de infravermelho utilizado em sistemas de controles remotos. Contendo um circuito interno que trate a onda infravermelha e que emita sinais de tensão pelo pino de sensoriamento ou de saída. Para que isso ocorra ele contém um pré-amplificador de ondas para ampliar o sinal recebido do diodo emissor do infravermelho.

1. Módulo Emissor TIL32

O módulo emissor TIL32 refere-se a uma LED que irá emitir as ondas de infravermelho. O diodo LED de infravermelho emite somente o sinal em uma frequência funcional de 32 kHz em uma tensão de 5v.

De acordo com Braga (2013), esse módulo é apenas um diodo emissor de infravermelho, com o alcance de 90 cm. Pois diodos de ondas infravermelhas podem ser configurados para emitir ondas de frequência muito altas

1. Módulo ESP8266

O módulo *wireless* ESP8266 é referente ao modelo 01, em que foi desenvolvido para poder conectar o microcontrolador a uma rede Wi-Fi de forma fácil, eficaz e por um baixo custo. Contém diversos modelos para implementação, mas a funcionalidade básica exercida por esse módulo é a conexão à rede Wi-Fi.

Ele suporta as redes 802.11 b/g/n, podendo trabalhar como Ponto de Acesso ou Estação de rede, enviando e recebendo dados.

A comunicação do módulo com o Arduino pode ser feita via serial pelos pinos de dados RX e TX ou se for necessário para facilitar a conexão utilizar um conversor digital para ter mais praticidade utilizando a placa de prototipação.

Espressif (2018) comprova, os detalhes do módulo e o suporte a redes Wi-Fi, protocolos e outros tipos de acesso à *internet*. E afirma a alta eficiência do módulo para conexões de redes Wi-Fi.

1. NodeMCU

O NodeMCU é uma placa de prototipação integrada com um módulo de Wi-Fi. A placa contém o módulo ESP8266 que processa e guarda os dados da programação e um conversor serial para ser conectado diretamente no computador pelo USB. É disponível utilizar a linguagem de programação C/C++ ou Lua que é menos difundida entre os sistemas de prototipagem.

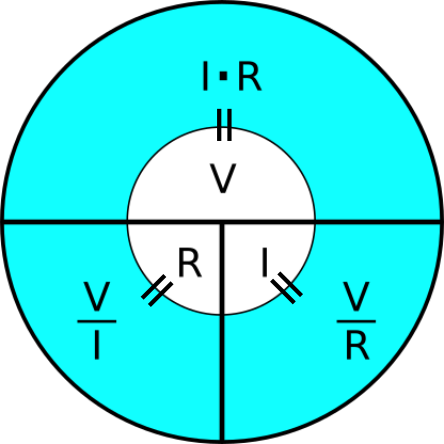
Segundo Oliveira (2017), o NodeMCU é um *firmware* baseado no ESP8266 que permite que seja programado em Lua e através da IDE do Arduino.

* 1. Lei de Ohm

Para desenvolver o esquema elétrico no Arduino e seus respectivos módulos é necessário ter algum entendimento sobre resistência elétrica, definida como Lei de Ohm. Para ligar o emissor e o receptor de infravermelho necessita de limitar a tensão para que não ocorra quais quer problemas de queima de equipamento ou curto-circuito.

De acordo com MARKUS (2002), a resistência elétrica é a oposição à passagem de corrente elétrica em um condutor. Essa oposição é causada pela dificuldade dos elétrons livres se movimentarem. A formula *V = R \* I*, representa que a intensidade da corrente (*I*) que depende da tensão (*V)* aplicada a resistência (*R)*. Utiliza-se a mesma formula para descobrir a corrente e a resistência de acordo com a figura 1.

1. – Lei de Ohm



**Fonte:** :<http://gt-mre.ufsc.br/moodle/course/view.php?id=2>

* 1. Rede

A utilização de redes entre dispositivos aumenta a disponibilidade e compartilhamento do acesso de dados, TORRES (2010) afirma que além de ter a facilidade de compartilhar esses dados há a vantagem de compartilhar recursos de máquinas como impressora ou o próprio acesso à internet auxiliando na redução de custos dos equipamentos e serviços.

De acordo com Comer (2007), as redes de computador usam uma variedade de meios de transmissão, incluindo as convencionais. Cada meio de transmissão e tecnologia apresenta vantagens, desvantagens e custos. Cada tipo de rede tem a sua necessidade para ser utilizado.

1. *Wireless*

Existem diversos tipos de redes sem fio como as redes via satélite, *bluetooth*, infravermelho. Essas redes têm em comum a comunicação de dados sem ter um meio físico, como cabos ou outros tipos de condutores.

De acordo com RUFINO (2011), é mais fácil para um iniciante a montagem, instalação e a configuração das redes Wi-Fi do que as outras tecnologias de redes cabeadas, não necessitando do entendimento muito técnico.

O padrão de comunicação da rede *wireless* é relativo ao 802.11, definido pelo IEEE o padrão de rede sem fio é conhecido como Wi-Fi, a LAN denominada como *Local Area Network* ou rede local sem fio opera com a frequência base de 2,4 GHz. Cada versão desenvolvida tem uma série de especificações como a velocidade a banda de frequência e a definição de comunicação.

TORRES (2010) afirma que o padrão 802.11 é usado em redes sem fio, em que a transmissão é por ondas de rádio. E que o alcance, a taxa de frequência e de transferência é determinada pelo padrão utilizado na camada de rede. Exemplo: 802.11b, 802.11g, etc.

Ao invés de transmitir sinais a partir de cabos, o ar serve como meio de transmissão de sinais via rádio, sendo de modo compartilhado, ou seja, é configurado para operar na mesma frequência.

1. Protocolos

Cada protocolo utilizado em uma rede tem uma função diferente, em uma conexão básica entre dois computadores são utilizados vários protocolos.

Protocolos como TCP/IP fornecem as regras para a comunicação. Eles contêm os detalhes de formatos de mensagens, descrevem o que um computador faz ao receber uma mensagem e especificam como o computador trata os erros ou outras condições anormais. O mais importante é que esses protocolos permitem que haja a comunicação através do computador, independente do *hardware* da rede de qualquer fornecedor em particular. (COMER, 2007).

Segundo TORRES (2010), para que duas máquinas possam se conectar e compartilhar dados é necessário que as duas utilizem a mesma arquitetura. Os protocolos auxiliam na conexão e compartilhamento de dados.

* 2. Sistema *web*

Um Sistema *Web* consiste em uma aplicação que é executada diretamente ou por meio proveniente de comunicação entre a *internet*, o usuário acessa a aplicação utilizando o navegador ou outro tipo de aplicação que interprete o HTML, sendo assim é uma aplicação que pode ser utilizada por qualquer dispositivo que tenha suporte e acesso à *internet.*

Para desenvolver um sistema Web é necessário utilizar algumas linguagens e metodologias para o desenvolvimento.

Para interface do sistema deverá ser desenvolvida em HTML e linguagens de *script* para que não comprometa as funcionalidades e que auxilie no visual atrativo com CSS.

Um sistema *de* acordo com Gama (2018) é um *site* hospedado em um servidor na *internet* onde o usuário cadastrado pode acessar as funções através de um navegador.

* + 1. HTML

A HTML ou *Hypertext Markup Language* é uma linguagem de marcação utilizada para exibir o contexto de um site, atualmente é a linguagem mais utilizada no desenvolvimento *Web.*

De acordo com Grannell (2009), a HTMT tem um aspecto curioso, que é o fácil entendimento básico da linguagem, qualquer um com algum conhecimento em computação pode ser capaz de montar uma página básica apenas aprendendo algumas *tags*, e tem a flexibilidade de deixar o visual atraente quando é combinado com o CSS.

“A HTML (*Hypertext Markup Language)* é uma linguagem baseada em uma mais antiga e muito mais complexa chamada SGML (*Standart Generalized Markup Language*). Por esse motivo, uma das maiores características da linguagem HTML é não ter uma estrutura rígida e exata. Ela apenas define a estrutura de uma página estabelecendo o que é título, texto, lista, subtítulo, local das imagens etc.” (MARCONDES, 2009, p. 40).

* + 1. CSS

O CSS ou *Cascading Style Sheets*, é uma linguagem de folha de estilos, com o intuito de tornar uma página na *web* apresentável.

Aprofundando um pouco em CSS, Segundo Grannell (2009), O HTML foi concebido como uma linguagem de *markup* estrutural, o conceito do CSS era simples, e mesmo assim revolucionário, o conceito era eliminar a apresentação e separar o desenho do conteúdo, deixando o HTML tratar da estrutura e o CSS para aplicar a apresentação visual.

O CSS tem a finalidade de organizar o HTML, Oliveira (2013), afirma que ao utilizar o CSS em um site ele altera os elementos do HTML de acordo com o padrão utilizado na construção do código, sendo definido o que é mostrado no navegador e padronizando o layout e o escopo das páginas.

* + 1. PHP

O PHP é uma linguagem de programação para o desenvolvimento de sistemas dinâmicos e interativos. De acordo com Soares (2010, p. 28) PHP é uma linguagem de programação poderosa, sendo mundialmente utilizada, principalmente em sistemas *Web* com a capacidade de manipular o HTML para desenvolver páginas dinâmicas.

De acordo com Davis e Phillips (2008), o PHP gera páginas web interativas no servidor onde é aplicado, ao contrário do HTML, em que o navegador utiliza etiquetas e *markups* para gerar uma página. O PHP circula entre a página requisitada e o servidor modificando a saída no HTML.

* + 1. MySql

O MySql é um Sistema de Gerenciamento de Banco de Dados relacional gratuito para o armazenamento de dados, sendo confiável e de flexível utilização. Sendo possível utilizar em sistemas básicos ou complexos pelas funcionalidades atribuídas.

Segundo Minorello (2010), o MySql é um servidor de banco de dados *multiuser* e *multitrhead* que utiliza o SQL, linguagem padronizada para manipular os dados em diversas bases de dados.

* 1. C++

O C/C++ é a linguagem padrão de programação utilizada para utilizar no Arduino, contendo diversas bibliotecas e utilitários da linguagem que auxilia no controle de entradas e saídas do sistema.

Silveira (2012) afirma, que a estrutura da linguagem de programação do Arduino é bastante simples, sendo formada por blocos de funções escritos em C/C++. Essas funções são definidas como sub-rotinas, para montar o programa principal.

* 1. Trabalhos correlatos

Nesse tópico serão apresentados os trabalhos relativos ao controle de vagas de estacionamento e a automatização utilizando o Arduino, sendo necessário para citar os problemas relatados, obter informações e para verificar os êxitos e problemas relatados.

* + 1. Sistema online para estacionamento de veículos

O sistema online para estacionamento de veículos foi desenvolvido para facilitar determinadas tarefas relacionadas a agilidade e facilidade da escolha de um local para estacionar. Utiliza-se a aplicação desenvolvida em Android para analisar e verificar um local de preferência com a possibilidade de estacionar no local desejado.

* + 1. Controle centralizado de equipamentos de ar-condicionado

Esse sistema de controle utiliza o Arduino para substituir o controle remoto de infravermelho e faz a conexão do *hardware* com o módulo de rede ZigBee que utiliza o mesmo padrão de rede do módulo ESP8266. Esse projeto auxilia no entendimento da conexão do sistema utilizando o Arduino e o servidor.

De acordo com CUNHA (2013, p. 13) “O protótipo deveria ser capaz de transmitir e receber dados de um servidor, onde uma aplicação web tornaria a interação do usuário com os aparelhos amigável”.

1. Metodologia

Esse trabalho tem o objetivo de desenvolver um sistema que facilite o controle do estacionamento com dispositivos móveis a partir da plataforma Arduino. A partir desse capitulo serão apresentadas as ferramentas e a metodologia necessária para o desenvolvimento.

* 1. Ferramentas

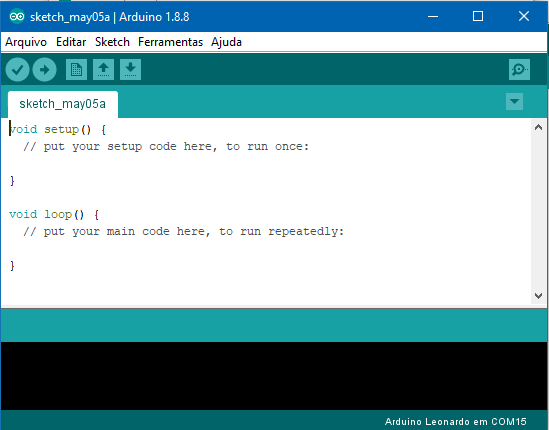
Para a programação no Arduino será necessário utilizar a própria IDE disponibilizada em seu respectivo site ou a alternativa para programar diretamente os microcontroladores da Atmel utilizando o Atmel *Studio* tendo suporte de programação para *Assembly*. A plataforma Fritzing será utilizada para facilitar a elaboração do esquema elétrico do projeto.

* + 1. Arduino IDE

Para programar o Arduino é necessário conter a IDE do mesmo. A aplicação disponibiliza o ambiente necessário para a programação, o *sketch* é a linguagem de programação reservada para o *upload* no Arduino, através da comunicação serial da placa. Após a compilação e o envio do *sketch* para a placa de prototipagem, ele fica responsável por controlar o I/O e o gerenciamento do sistema.

O método de compilação do *sketch* traduz a linguagem de programação C/C++ para a linguagem de máquina para que o microcontrolador possa interpretar. Na aplicação há o processo de *upload* da programação para o Arduino, em que, após executar a ação o Arduino não necessitará a conexão direta com o computador para o funcionamento, apenas de uma fonte de energia que supra a necessidade para continuar funcionando. Como exemplo na figura 2 a imagem da tela inicial do Arduino IDE.

1. – Tela Inicial do Arduino IDE

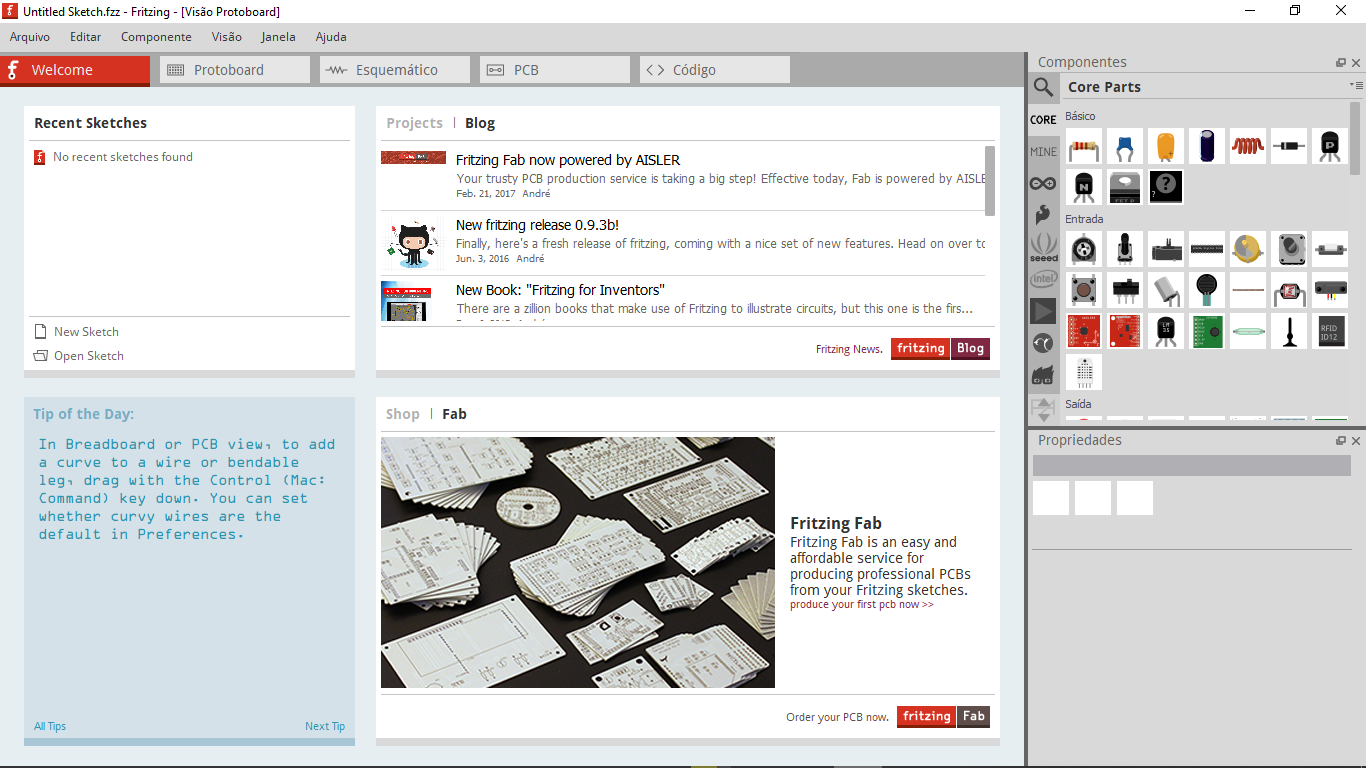


Fonte: Autores

* + 1. Fritzing

O Fritzing é um *software* para a criação de esquemas, diagramas, prototipação e de *layouts* para placas de circuito impresso. Alguns modelos de placa de prototipação podem ser encontrados na aplicação como o Arduino ou *Raspberry*-pi. De Acordo com a figura 3 é exibido a tela inicial da aplicação.

1. – Tela inicial Fritzing

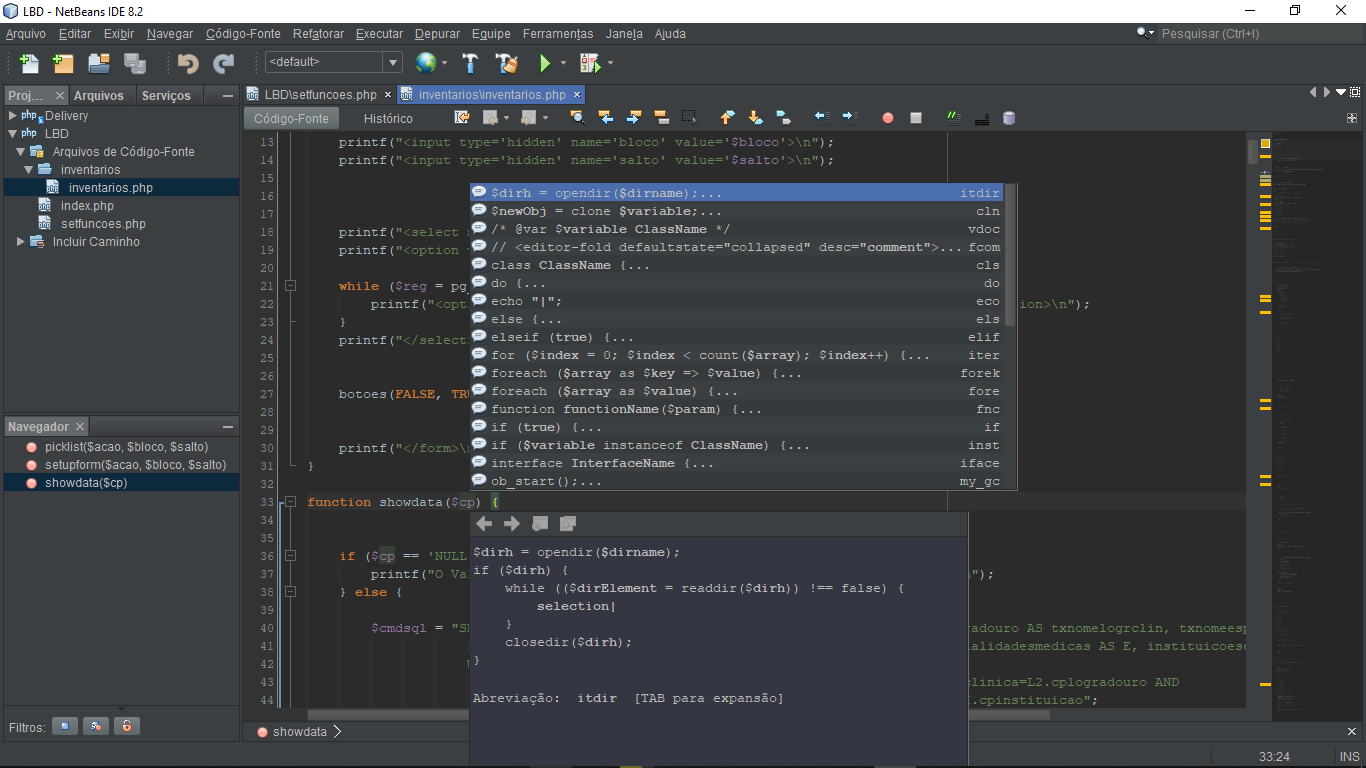


**Fonte:** Autores

* + 1. NetBeans

Esse *software* é destinado para o desenvolvimento de sistemas em diversas linguagens, a distribuição direcionada para PHP permite a edição de HTML, CSS, JavaScript, PHP e outros tipos de linguagens para o desenvolvimento de aplicações *Web*. Esse ambiente de desenvolvimento facilita o entendimento do código com dicas e preenchimento automático na programação, na separação das funcionalidades do sistema a ser desenvolvido e com validação para implementar de maneira ágil representado na imagem 4.

1. – Tela inicial NetBeans



**Fonte:** Autores

* + 1. Kit Básico do Arduino

O kit básico do Arduino consiste em alguns componentes:

*Jumpers* – Para conectar os pinos e os cabos utilizados no projeto.

Placa de prototipação – Arduino ou NodeMCU.

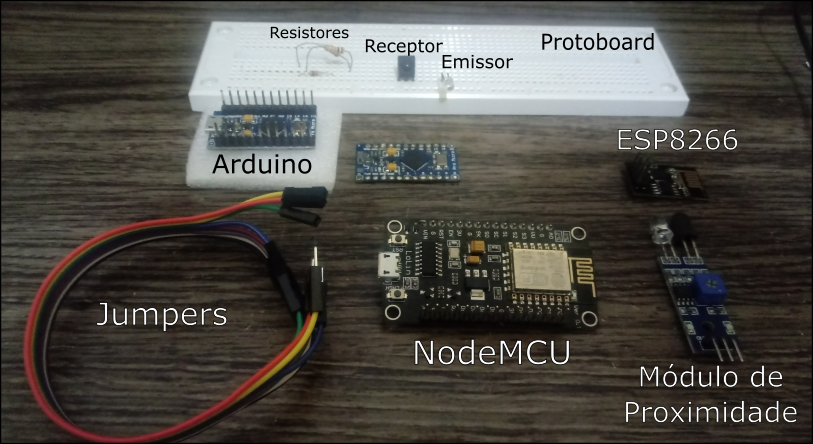
*Protoboard* – Para prototipação em uma base de fácil manuseio sem necessitar de um circuito impresso.

Resistores – Para limitar a tensão de acordo com a necessidade e com o auxílio da formula básica da Lei de Ohm.

Módulos – Exercem funções essenciais para agregar ao projeto como sensores de proximidade, módulos para a conexão em redes, controladores de motor, etc.

A figura 5 é um exemplo do kit Arduino dos integrantes do grupo.

1. - Kit Arduino Básico

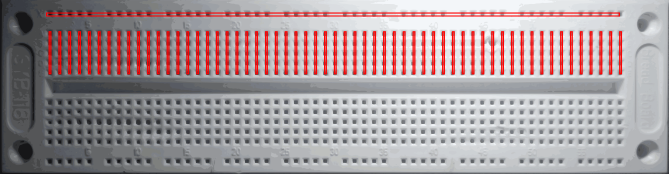


**Fonte:** Autores

* + 1. *Protoboard*

A *protoboard* éuma placa para o auxílio na prototipação que facilita a conexão de componentes eletrônicos. É muito utilizada na área de eletrônica e de automação pois não necessita de soldar os componentes para haver a condução. Cada tipo e tamanho da placa contém um padrão de conexão, mas o padrão mais comum é representado na figura 6 com os traçados em vermelho compõem os contatos da placa.

1. - *Protoboard*



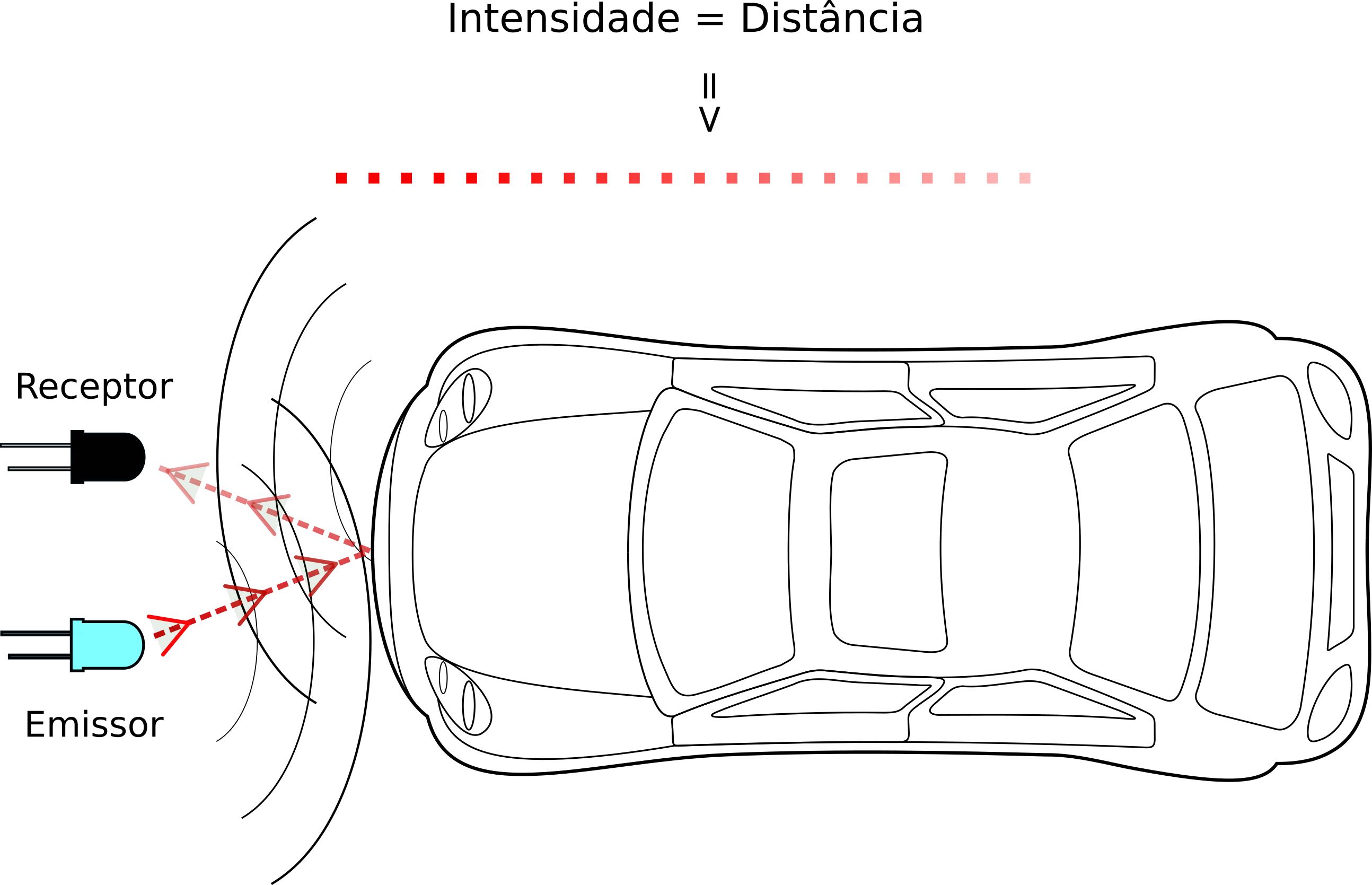
**Fonte:** Autores

* + 1. Sensor de proximidade

O sensor de proximidade infravermelho contém o emissor e o receptor de frequência infravermelha, funcionando de maneira que o emissor envie as ondas de luz e que o objeto mais próximo reflita para o receptor. De acordo com determinada intensidade da recepção dessas ondas pode-se definir a distância do objeto referente a imagem 7.

Segundo Filipeflop(2015), o sensor de infravermelho possui um circuito com o emissor e receptor de infravermelho, posicionados um ao lado do outro. Quando um objeto passa em frente ao emissor, o infravermelho é refletido e detectado pelo receptor.

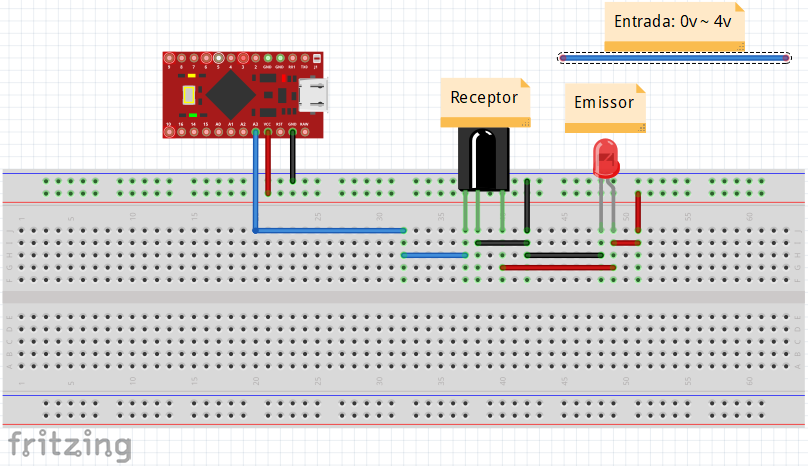
1. – Sensor de proximidade de infravermelho



**Fonte:** Autores

O sensor de proximidade pode ser desenvolvido com um esquema simples, o funcionamento é parecido com o módulo, mas retorna uma tensão variante para a placa referindo a distância, sendo necessário alterar a programação interna da placa de prototipação que irá o implanta-lo. O circuito apresentado na imagem 8 representa o esquema do circuito.

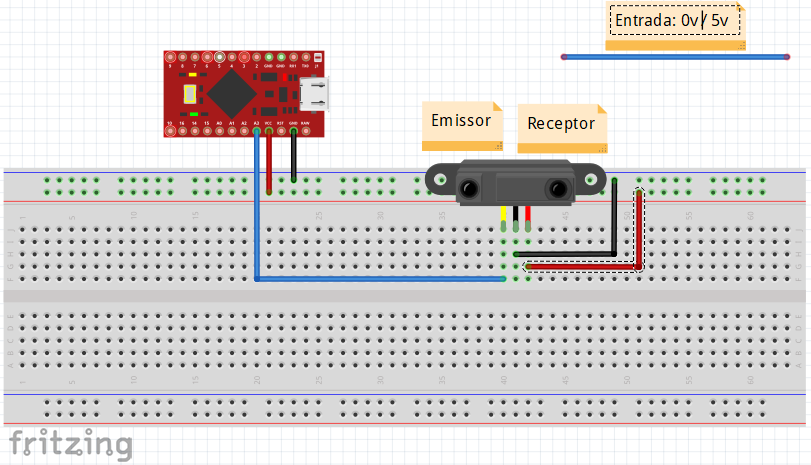
1. –Circuito do sensor de proximidade (Arduino)



**Fonte:** Autores

O módulo de sensoriamento de proximidade de infravermelho tem o funcionamento mais simples, pois para regular a distância de percepção de um objeto é necessário regular fisicamente no mesmo, funcionando de maneira que o objeto ao se aproximar do campo de reflexão, retorna a saída de dados 5v de tensão e 0v se não estiver próximo, representado na figura 9. O privilégio de utiliza-lo é a precisão e o alcance.

1. –Circuito módulo do sensor de proximidade (Arduino)



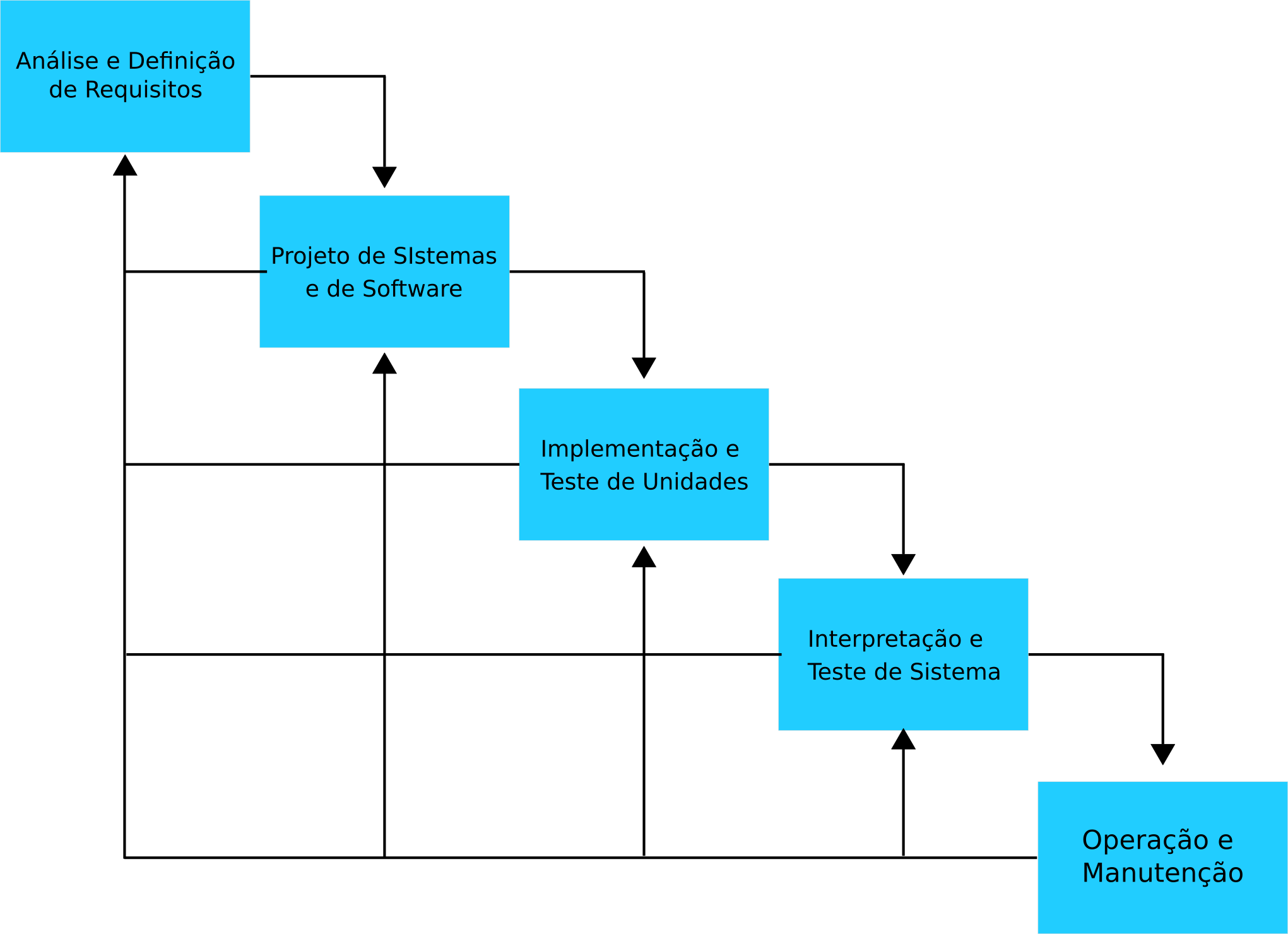
**Fonte:** Autores

* 1. Procedimentos

O desenvolvimento do sistema será realizado utilizando o modelo cascata, em que Sommerville (2012) justifica que esse modelo é denominado cascata pelo fator de migração de uma etapa para a outra, mas pode ser chamado de ciclo de vida de software.

As principais etapas são a análise e definição dos requisitos, o projeto de sistemas de software, implementação e teste de unidades, a integração e testes de sistema e a operação de manutenção. De acordo com a figura 10.

1. – Modelo Cascata



Fonte: <http://jkolb.com.br/o-modelo-em-cascata/>

Este processo de desenvolvimento foi escolhido devido a fatores e características do sistema que foram analisados, como tempo de desenvolvimento, complexidade do software e disponibilidade do cliente.

O levantamento dos requisitos é uma etapa crucial para o desenvolvimento do sistema, onde se define as regras de negócio, do sistema e de *software*. O documento de requisitos referente ao apêndice A.

Após levantar todos os requisitos e definir as regras, é necessário desenvolver o diagrama de caso de uso que se encontra no apêndice B representando os atores que irão interagir com o sistema e as respectivas funcionalidades disponíveis para cada atividade.

* 1. Desenvolvimento

A seguir será apresentado as etapas necessárias para o desenvolvimento do projeto utilizando o modelo cascata como base definindo as etapas concluídas.

* + 1. Análise e Definição de Requisitos

É definido as funções, restrições e os objetivos do sistema. De acordo com análises e consultas com o usuário do sistema é efetuado o levantamento desses requisitos e o detalhamento.

Foram levantados os requisitos do sistema para descrever as funcionalidades de acordo com o apêndice A. Relatando que o usuário do sistema é a empresa que irá usufruir as funcionalidades do sistema.

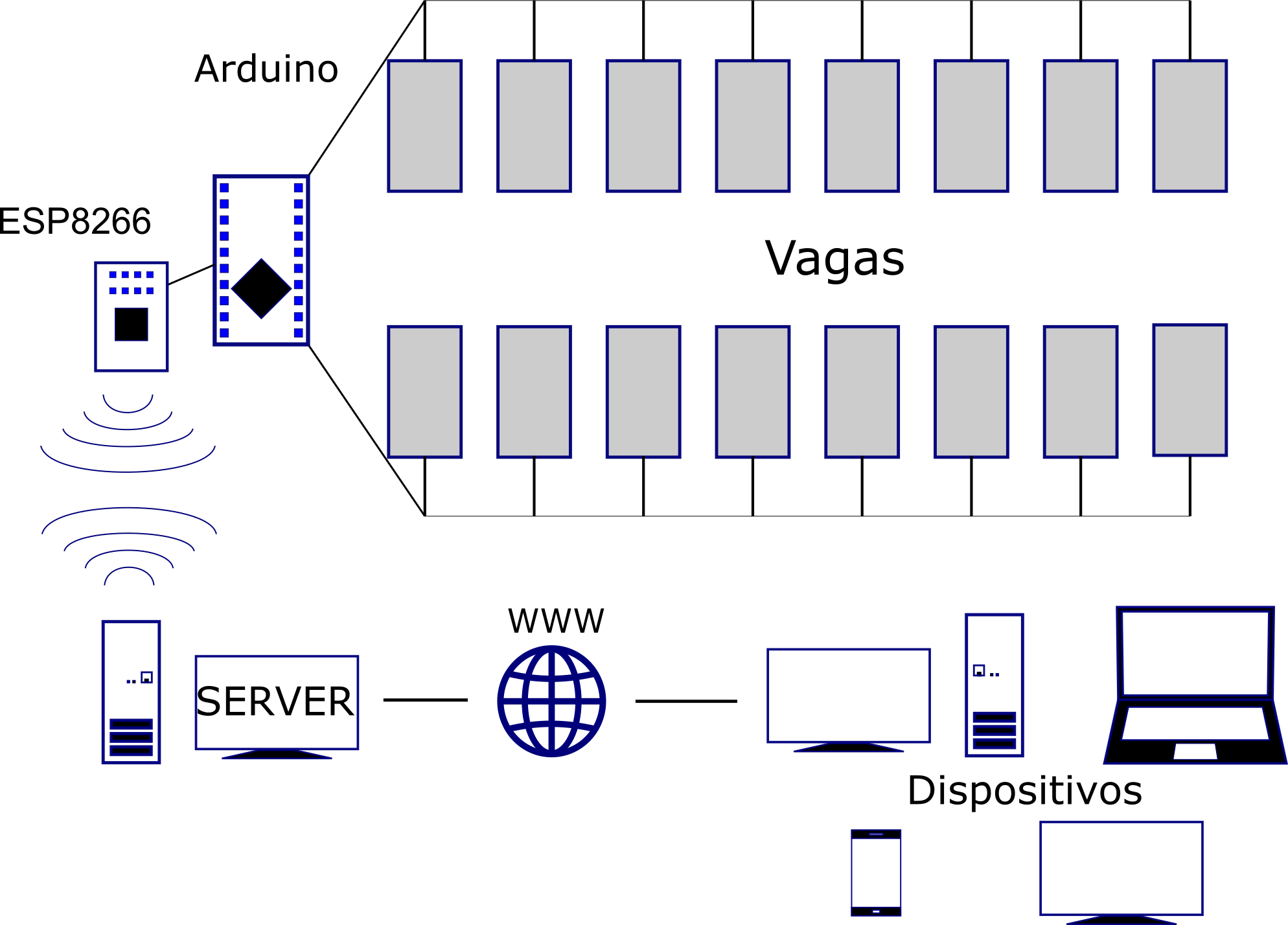
* + 1. Projeto de Sistema e de *Software*

Essa etapa organiza os requisitos do *software* e do *hardware* do sistema, definindo toda a arquitetura para o desenvolvimento. Permite identificar e descrever as abstrações fundamentais do software e suas relações. Após o levantamento dos requisitos foi modelado o diagrama de Caso de uso referentes ao apêndice B, definindo as funções e o usuários declarados como atores para haver uma representação gráfica do *software.*

E o desenvolvimento do diagrame de Classe referente ao apêndice C, representando as classes utilizadas no sistema e as classes que utilizarão o modelo DAO (Classes de criação de objetos para aceso de dados).

Foi definido todas as ferramentas necessárias para o desenvolvimento, pois o projeto consiste em uma arquitetura específica para a implementação, como a utilização do Arduino, módulos de conexão de rede e sensores. Na imagem 11 é apresentado a maneira de como será o funcionamento do projeto. Entretanto é possível realizar o projeto de duas maneiras.

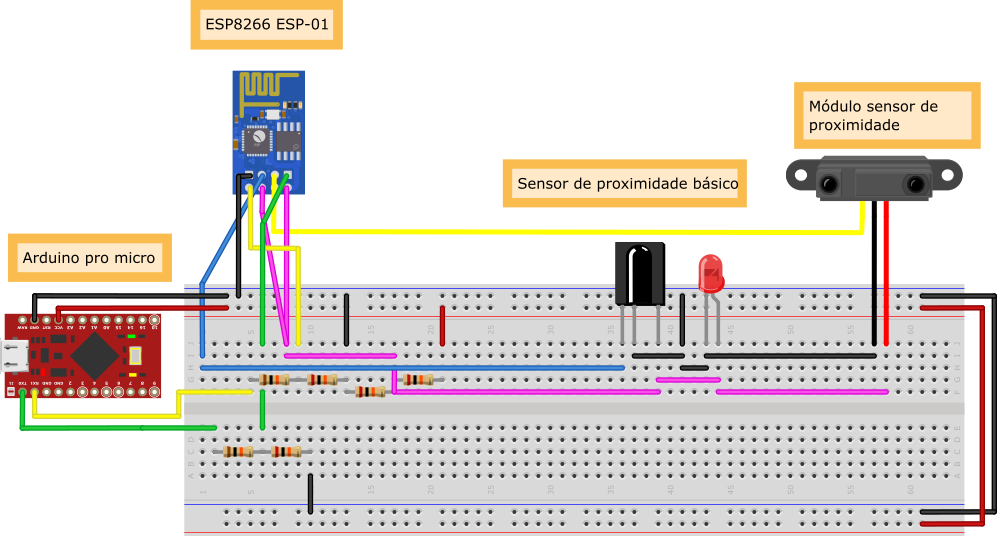
1. – Modelo de funcionamento



Fonte: Autores

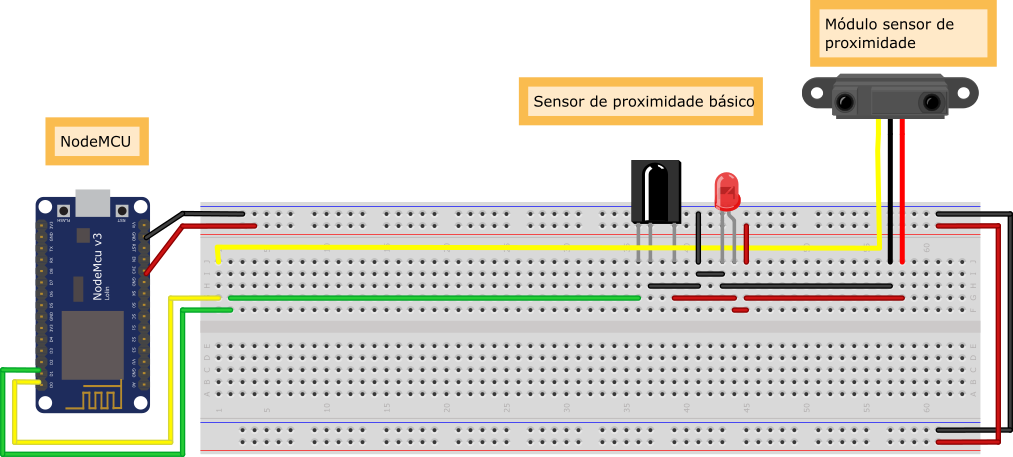
O projeto pode utilizar o Arduino para controlar a porta serial do ESP8266 para gravar a programação ou pode-se utilizar o NodeMCU não sendo necessário de outros módulos para comunicação wireless. Ambos os esquemas estão representados sucessivamente na figura 12 e 13.

1. – Circuito completo (Arduino)



Fonte: Autores

1. – Circuito completo (NodeMCU)



Fonte: Autores

* + 1. Implementação e Teste de Unidades

É o desenvolvimento do sistema como um conjunto de programas ou unidades, cada unidade deve haver a própria função.

A aplicação foi desenvolvida em classes referentes ao apêndice C.

* + 1. Integração e Testes de Sistemas

As unidades desenvolvidas são integradas em um sistema como um todo. Sendo implementado e testado completamente.

Essa etapa foi desenvolvida unindo todos os módulos internos e externos do sistema para que possa ser executado o sistema completo.

* + 1. Manutenção

Essa etapa tem o objetivo de garantir a correção de erros após a finalização de todas as outras etapas melhorando a implementação. Pois essa etapa é implementada após a finalização do sistema para prevenir e corrigir possíveis erros e inconsistências do sistema.

1. Cronograma

Este capítulo mostra o cronograma do trabalho a ser desenvolvido.

1. - Cronograma

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Etapas | 2018 | | | | |
| AGO | SET | OUT | NOV | DES |
| Levantamento do material e instrumentos | X |  |  |  |  |
| Início do desenvolvimento do TG | X |  |  |  |  |
| Revisão da literatura | X | X | X |  |  |
| Entrega da 1ª versão ao orientador |  | X |  |  |  |
| Revisão do orientador |  | X |  |  |  |
| Entrega da 2ª versão ao orientador |  |  | X |  |  |
| Revisão do orientador |  |  | X |  |  |
| Revisão e alteração do Documento de requisitos |  |  | X |  |  |
| Finalização do documento |  |  |  | X |  |
| Entrega das cópias do TG |  |  |  | X |  |
| Desenvolvimento da Apresentação |  |  |  | X | X |
| Qualificação |  |  |  |  | X |

**Fonte:** Autores

**Referência**

COMER, Douglas E. **Redes de computadores e internet.** 4. ed.Porto Alegre: Bookman, 2007.

CUNHA, Thiago F. **Controle centralizado de equipamentos de ar condicionado via rede sem fio ZigBee.** 2013. Disponível em: <https://wiki.sj.ifsc.edu.br/wiki/images/1/14/TCC\_ThiagoFelipeCunha.pdf>. Acesso em: 20 jun. 2018.

ESPRESSIF System. ESP8266EX: Datasheet. 2018. Disponivel em: <https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\_datasheet\_en.pdf.> Acessado em:12 nov 2018.

DAVIS, Michele E.; PHILLIPS, Jon A.Aprendendo **PHP e MySQL**. Rio de Janeiro: Alta Books. 2008.

FILIPEFLOP. 2015. Disponível em: <https://www.filipeflop.com/blog/sensor-infravermelho-arduino/>. Acesso em: 11 nov. 2018.

GRANNELL, Craig. **O Guia Essencial de Web Desing com CSS e HTML**. Rio de Janeiro: Ciência Moderna Ltda, 2009.

JUNIOR, Wilson M. Lopes. Fluxo de Automóveis nos Estacionamentos Públicos e Privados da Cidade.Turística de Angra dos Reis-RJ. **Turismo em Análise**. Angra dos Reis**,** v. 27, n. 2, p. 433 – 442, jun. 2016. Disponível em: <http://dx.doi.org/10.11606/issn.1984-4867.v27i2p429-453>. Acessado em:11 nov. 2018.

MARCONDES, Christian A. **HTML 4.0 Fundamental**: A Base da Programação para WEB. 2. ed. São Paulo: Érica, 2007.

MARKUS, Otávio. **Circuitos Elétricos:** Corrente contínua e Corrente Alternada. 9 ed. São Paulo: Érica, 2011.

MCROBERTS, Michael.**Arduino básico**.São Paulo: Novatec. 2015. 2 ed.

Acesso em:13 nov. 2018.

MINORELLO, Danilo; VARGAS, Elton..**PHP/Mysql**: Web.viena 2010. 2 ed.santa cruz do rio pardo.

MONK, Simon. **Programação com Arduino**: Começando com sketches. 2 ed. Porto Alegre: Bookman, 2017.

PEREIRA, Fábio. **Microcontroladores PIC**:Técnicas avançadas. São Paulo: Érica, 2012.

RUFINO, N. M. O. **Segurança em Redes sem Fio.** 3. ed. São Paulo: Novatec, 2011

SILVEIRA, João A.Arduino: Cartilha para programação em C. **Revista do Arduino**. v. 1, n. 1, p.2, jan 2012. Disponivel em: < http://ordemnatural.com.br/pdf-files/CartilhadoArduino\_ed1.pdf>. Acesso em: 12 nov. 2018

SOMMERVILLE, Ian. **Engenharia de Software**.9 ed São Paulo: Pearson Addison Wesley,.2012.

SOARES, Walace. **PHP 5**:Coceitos, Programação e Integração com Banco de Dados. 3 ed. São Paulo: Érica, 2010

TORRES, Gabriel. **Redes de computadores.** Rio de Janeiro: Novaterra, 2010.

TURIM, Fábio M.; LEONARDO, Guilherme de L.; ESPERANÇA, Lucas Augusto. **Sistema online para estacionamento de veículos**. 2018.Disponível em: <https://www.fatecourinhos.edu.br/academico/acervo/trabalho/1000536>

Acesso em: 18 out. 2018.

VISHAY. Photo Modules for PCM Remote Control Systems. 2013. Disponível em: <https://html.alldatasheet.com/html-pdf/26602/VISHAY/TSOP1830/182/1/TSOP1830

.html>.Acesso em: 13 nov 2018.

APÊNDICE A – Documento de Requisitos

**Principais envolvidos**

A seguir, é especificado os envolvidos no projeto como um todo, ou seja, engloba os usuários (proprietário e clientes do mesmo) e os desenvolvedores.

* **Usuários do Sistema**

Empresa: Gerente do sistema da empresa que irá fornecer as informações do estacionamento, como a base da planta, a quantidade de vagas, etc., sendo o único capaz de fazer alterações privilegiadas.

Usuário: Escolhe a vaga de sua preferência a partir das opções disponíveis no sistema.

* **Desenvolvedores do Sistema**

Alunos do curso de Análise e desenvolvimento de sistema da Fatec Ourinhos do período vespertino desenvolverão o sistema com o auxílio do respectivo orientador.

**Requisitos do Sistema**

Conforme os dados levantados referente a necessidade, funcionalidade e usabilidade, obteve-se os seguintes requisitos.

1. **Requisitos Funcionais**

Observações:

Os locais marcados com \* (asterisco) são definidos como campos de preenchimento obrigatório.

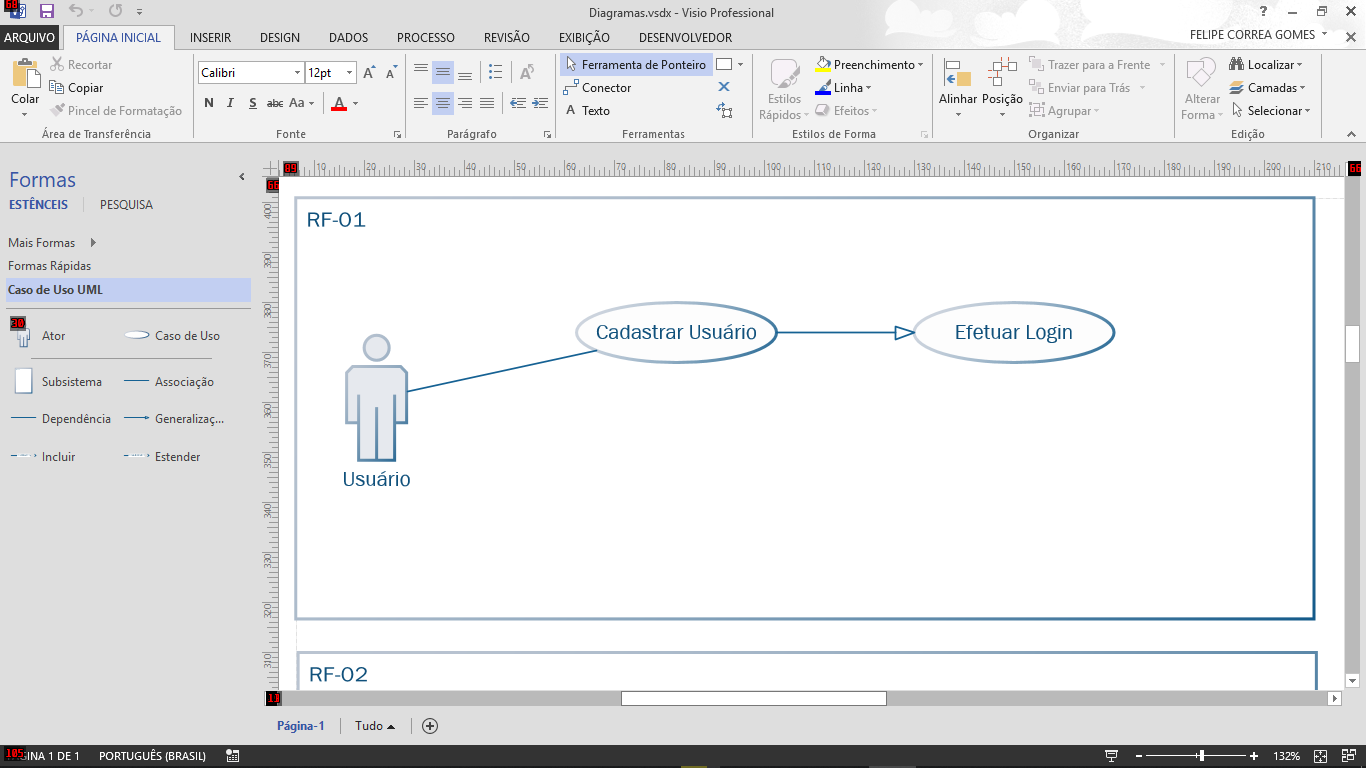
1. Cadastrar Usuário
   1. O sistema deve permitir incluir, excluir, alterar e consultar o cadastro do usuário.
   2. Para incluir os dados são necessários os seguintes campos: Nome\*, CPF\*, RG, Data de Nascimento\*, Endereço\*, Telefone\*(celular e/ou fixo), E-mail\*, senha\*, Veículo\*, Habilitação\* e Data de Cadastro\*.
      1. Para o preenchimento dos campos do endereço deverá conter o CEP\*, Rua\*, Número\*, Complemento, Bairro\*, Cidade\* e UF\* (sigla do estado).
      2. O campo de telefone deverá preencher obrigatoriamente ao menos um número.
      3. Os campos para Veículo serão utilizados para identificação sendo necessário preencher: Placa\*, Ano, Cor, Marca e Modelo.
      4. Os campos a serem preenchidos na Habilitação serão apenas o Número e a Categoria. O campo número é referente ao número de registro da habilitação do condutor.
      5. O campo de senha deverá ser preenchido no mínimo com 8 dígitos.
      6. A data de cadastro será gerada automaticamente pelo sistema.
   3. Após efetuar a inclusão do registro o usuário será requisitado automaticamente para efetuar o *login* (conforme RF - 02).
   4. O sistema permitirá alterar todos os campos exceto o CPF.
   5. O sistema permitirá a exclusão de todos os registros referentes ao solicitado
2. Efetuar *Login*
   1. O sistema deve permitir consultar o usuário cadastrado.
   2. Os campos que permitem realizar a conexão da sessão será: *Login*\* e Senha\*.
      1. O campo para login permitirá o preenchimento com CPF ou *E-mail* para a validação dos dados.
   3. Ao efetuar o *login* poderá ser possível reservar a vaga (conforme RF – 03).
3. Reservar Vaga
   1. O sistema deve permitir incluir, excluir e consultar a reserva de vaga.
   2. O sistema deve permitir o preenchimento dos seguintes campos para reservar a determinada vaga: horário de entrada\*, horário de saída\*, data de entrada\*, data de registro\*, vaga\* e usuário\*.
      1. O campo da data de registro será preenchido automaticamente após efetuar a ação de incluir.
      2. O campo da data de entrada é referente ao dia que o usuário irá ocupar a vaga.
      3. Para reservar a vaga é necessário selecionar a vaga disponível sendo tratada como campo e confirmar a ação preenchendo o horário de entrada e de saída.
      4. O usuário poderá realizar apenas uma reserva.
      5. Para selecionar a vaga disponível é necessário consultar os estados das vagas (de acordo com RF - 04) e o horário.
      6. Os campos do horário devem ser separados com o horário de entrada e o de saída, referentes ao horário que o usuário irá ocupar a vaga.
   3. O usuário terá a opção de cancelar o agendamento da vaga.
   4. Após o usuário reservar a vaga será alterado o estado de ocupação (de acordo com RF-04).
4. Gerenciar Vagas
   1. O sistema deverá permitir incluir, excluir, alterar, listar e consultar as vagas e seus respectivos estados.
   2. Os campos para inclusão das vagas são referentes ao estado\* e o número da vaga\*.
      1. Os estados da vaga variam entre Livre, Ocupado, Reservado e Minha Vaga.
         1. O estado Livre é determinado a vagas sem reservas ou ocupações.
         2. O estado ocupado é direcionado para vagas que após a reserva ou não esteja obrigatoriamente habitado por algum veículo.
         3. O estado Reservado é conduzido a vagas que tem horário determinado, mas não foram ocupas por quaisquer veículos.
         4. O estado Minha Vaga é uma condição genérica vinculado com reserva apenas para a exibição.
   3. O usuário que houver efetuado o login (conforme RF - 02) e reservado uma vaga (conforme RF-03), a vaga que coincide com a reserva do usuário será alterada para o estado minha vaga, para fins de exibição da vaga no qual ele reservou.
   4. Não é necessário estar cadastrado para visualizar as vagas e os respectivos estados.
5. Relatar Problema
   1. O sistema deve permitir apenas incluir e consultar a descrição do problema.
   2. Para a inclusão do problema é necessário o usuário preencher os seguintes campos: Motivo\*, Descrição\* e Data do Registro\*.
   3. Data do registro será gerada automaticamente.
   4. É necessário que o usuário tenha efetuado o login para relatar o problema.
6. Gerar Relatório
   1. O sistema deve permitir a criação de relatórios referente a utilização do estacionamento.
   2. O sistema deve solicitar a entrada por período (diário, semanal, mensal ou anual).
   3. O sistema deverá mostrar a frequência de usuários cadastrados e o do uso do estacionamento, o tempo de permanência, a quantidade de ocorrências e os utilizadores do sistema.
   4. O sistema deve totalizar o acesso dos usuários na plataforma.
   5. O sistema deve haver a opção de salvar o relatório em PDF ou de imprimir.
7. **Requisitos Não Funcionais**
8. Requisitos de Interface
9. A interface deve ser atraente e amigável tendo facilidade de permitir que os usuários se familiarizarem com ela de forma rápida para tirar o maior proveito possível.
10. Requisitos de Usabilidade
11. As mensagens de erro deverão ser objetivas, orientando os usuários a solucionar o problema e não impedindo o progresso do mesmo no sistema, pois comprometeria o funcionamento do sistema.
12. Requisitos de Segurança

APÊNDICE B – Diagrama de *Use Case*

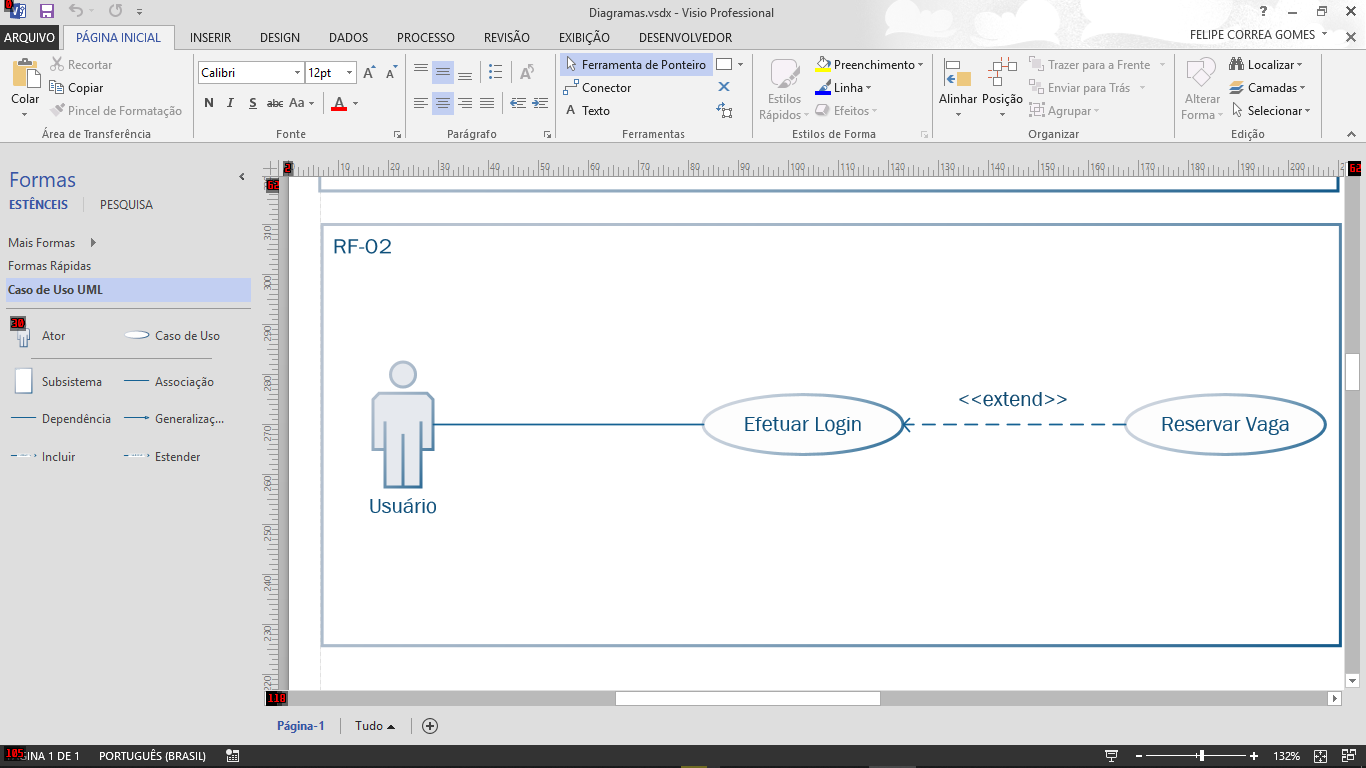
Este tópico apresenta as especificações dos requisitos do sistema de controle de vagas para estacionamento. A atividade de análise de requisitos foi conduzida aplicando-se técnicas de modelagem de casos de uso. O modelo apresentado foi elaborado usando UML a partir da ferramenta Visio.

**Modelo de *use case***

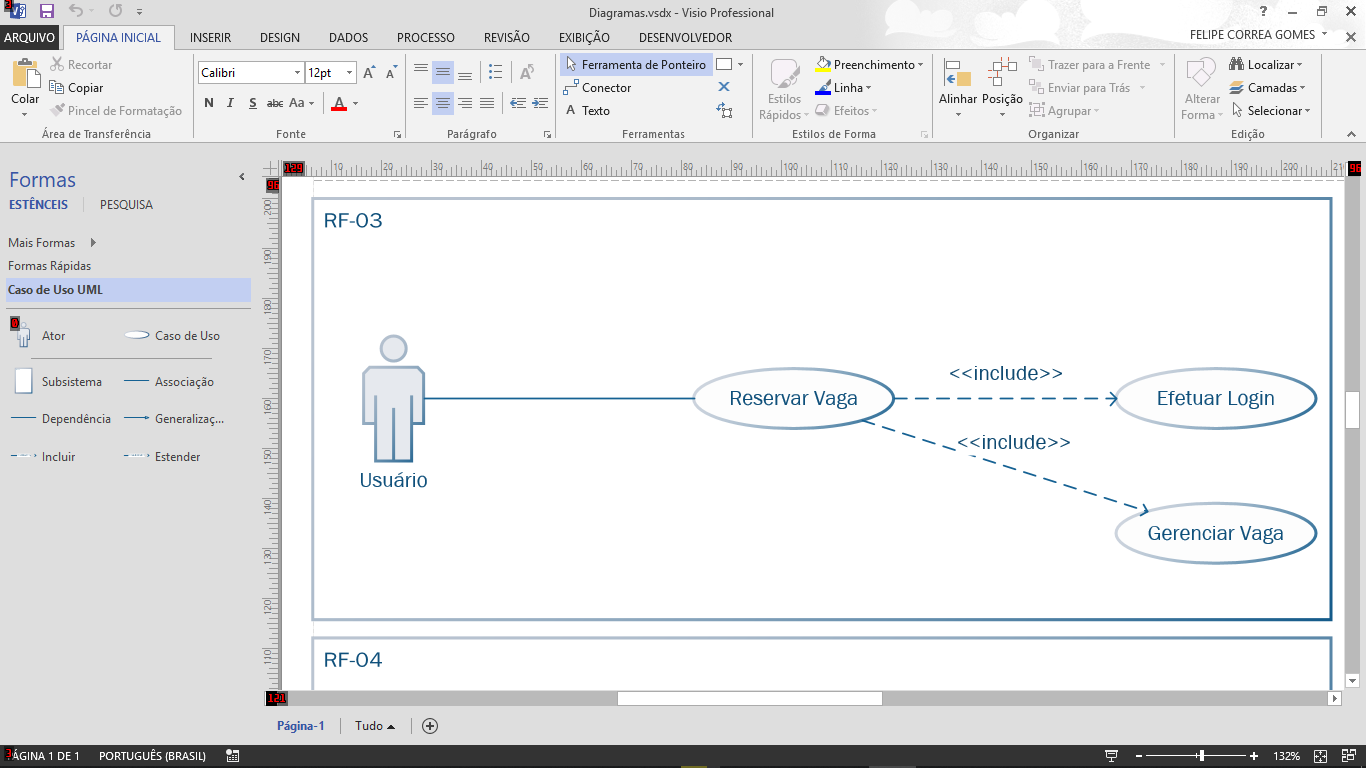
**RF- 01 –** Cadastrar Usuário



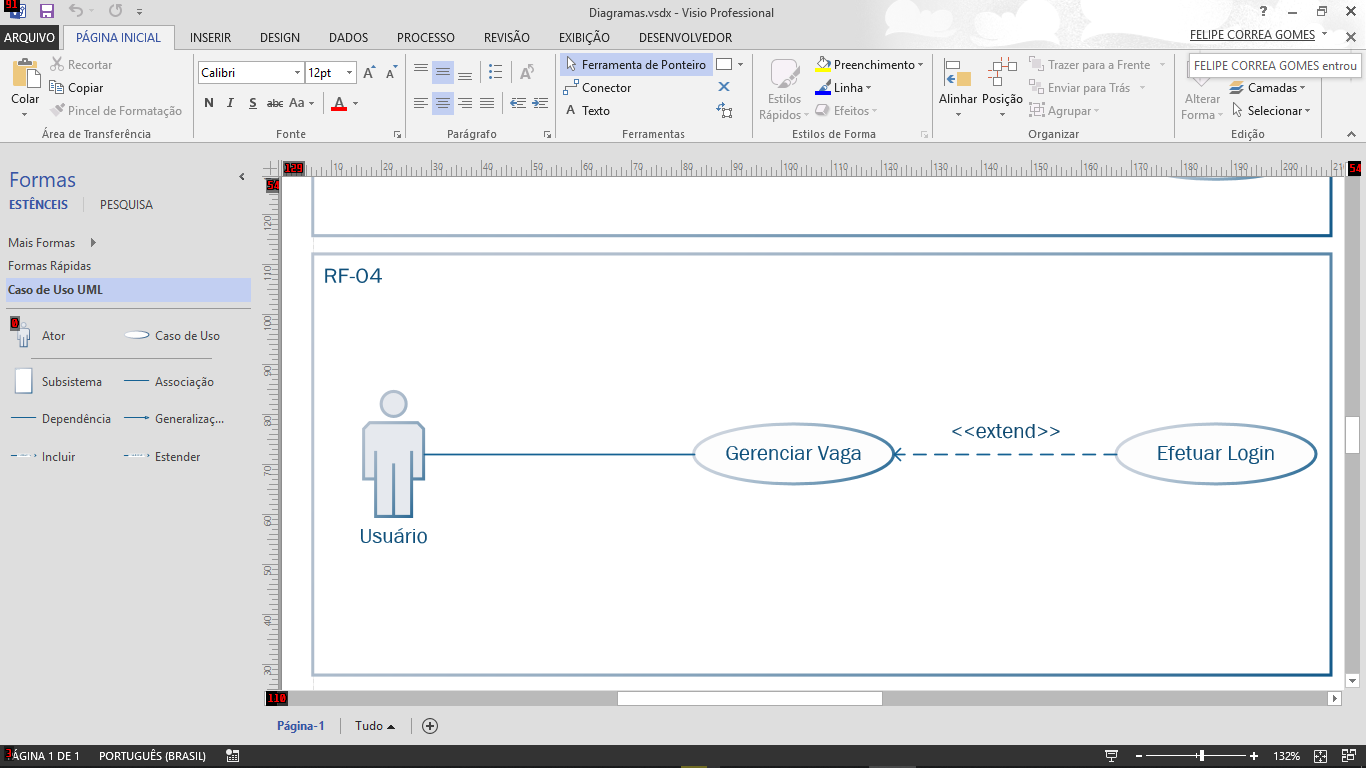
**RF- 02 –** Efetuar login



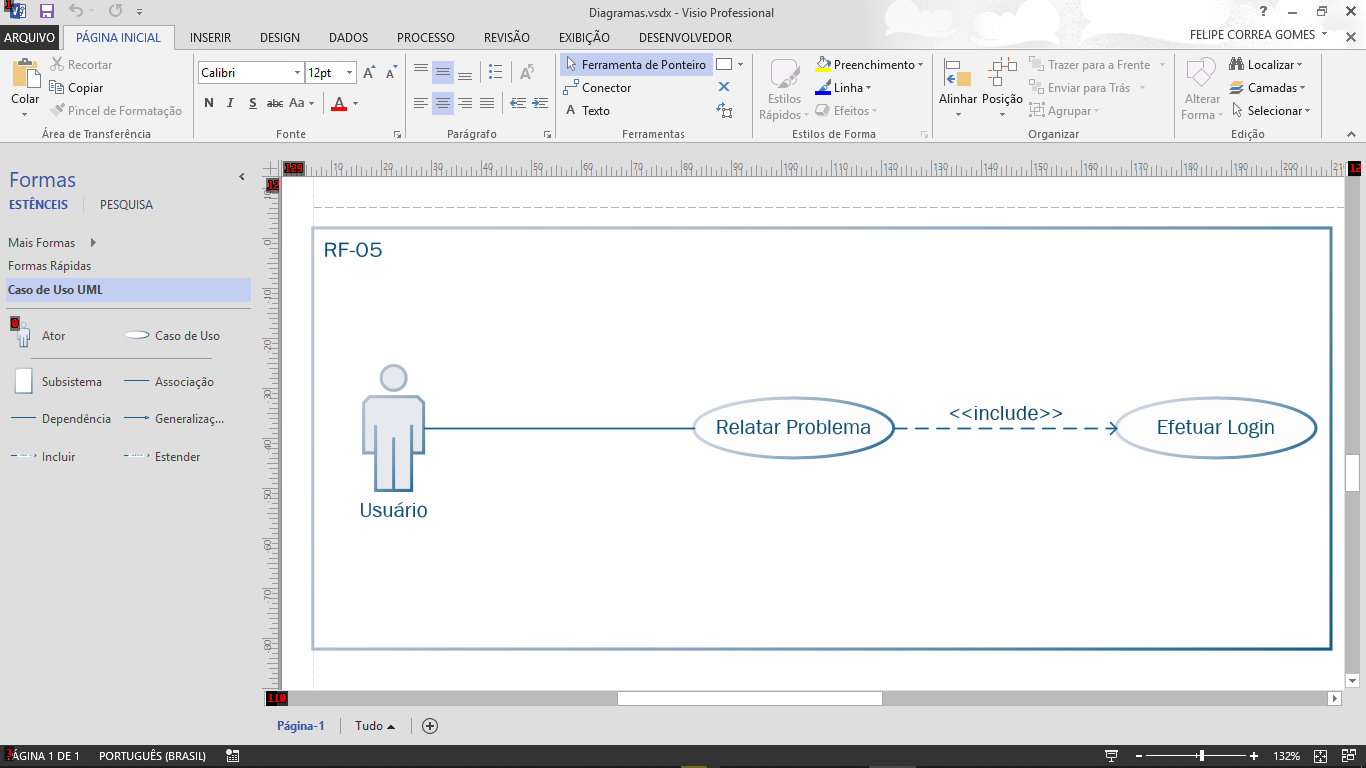
**RF- 03 –** Reservar Vaga



**RF- 04 –** Gerenciar Vaga

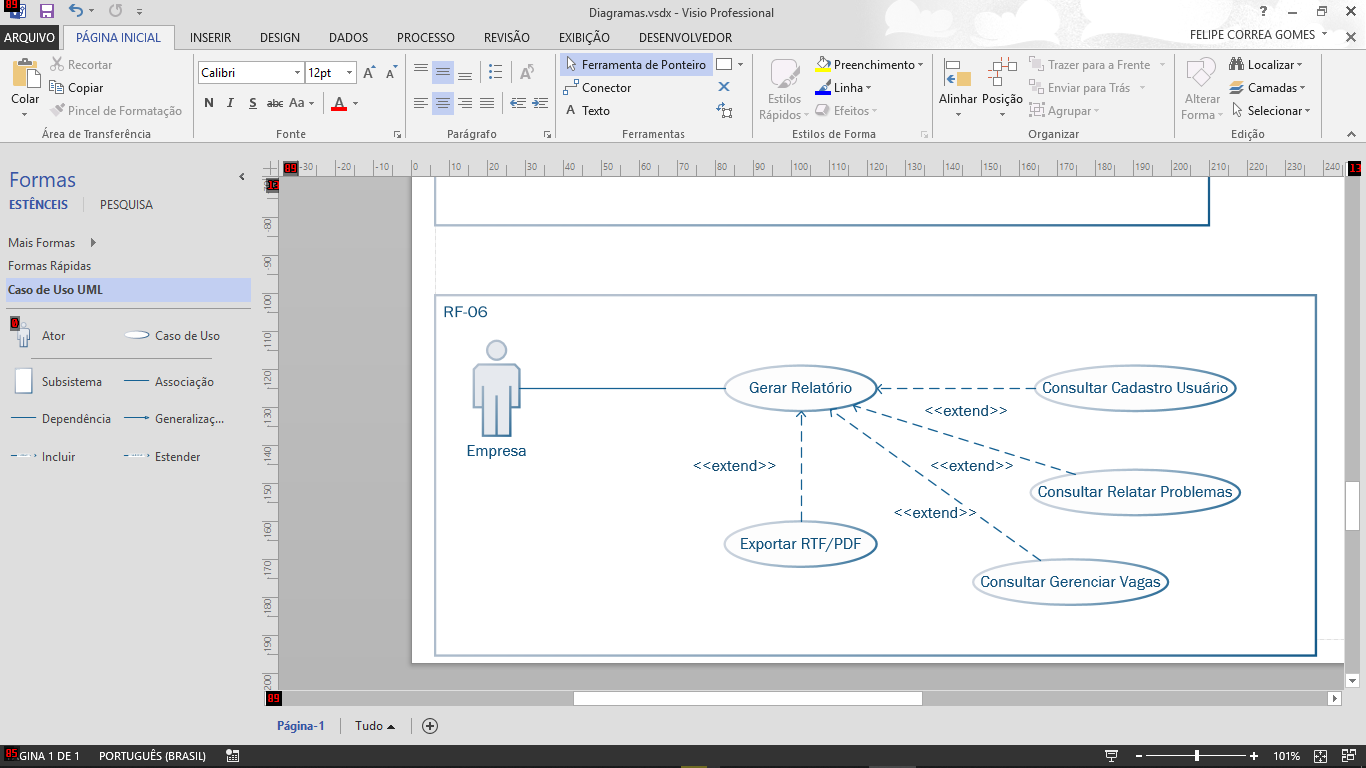


**RF- 05 –** Relatar Problema

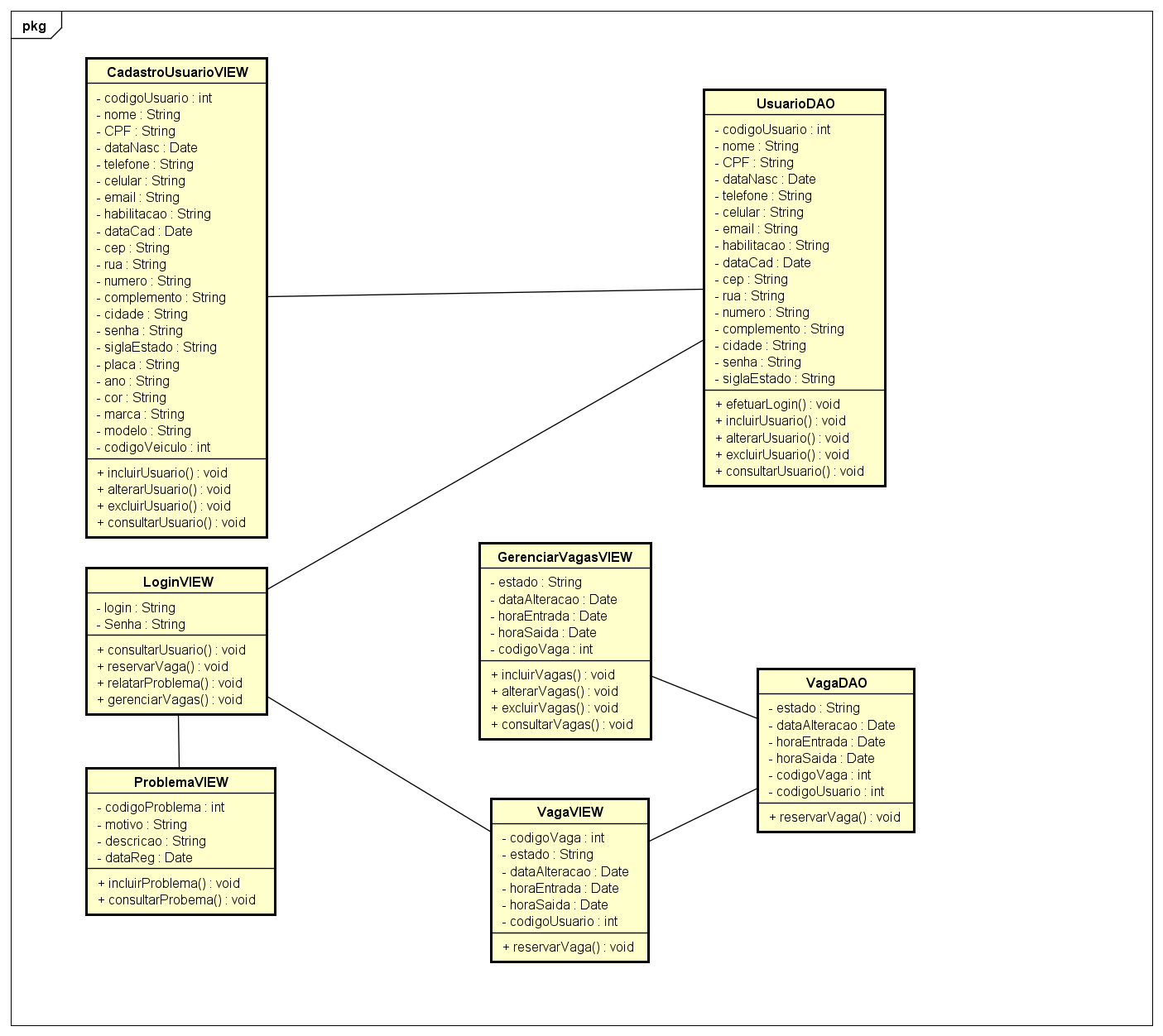


**RF- 06 –** Gerar Relatório

**APENDICE C –** Diagrama de classe



Esse tópico apresenta o diagrama de classe do sistema.



APENDICE D – Protótipos o sistema

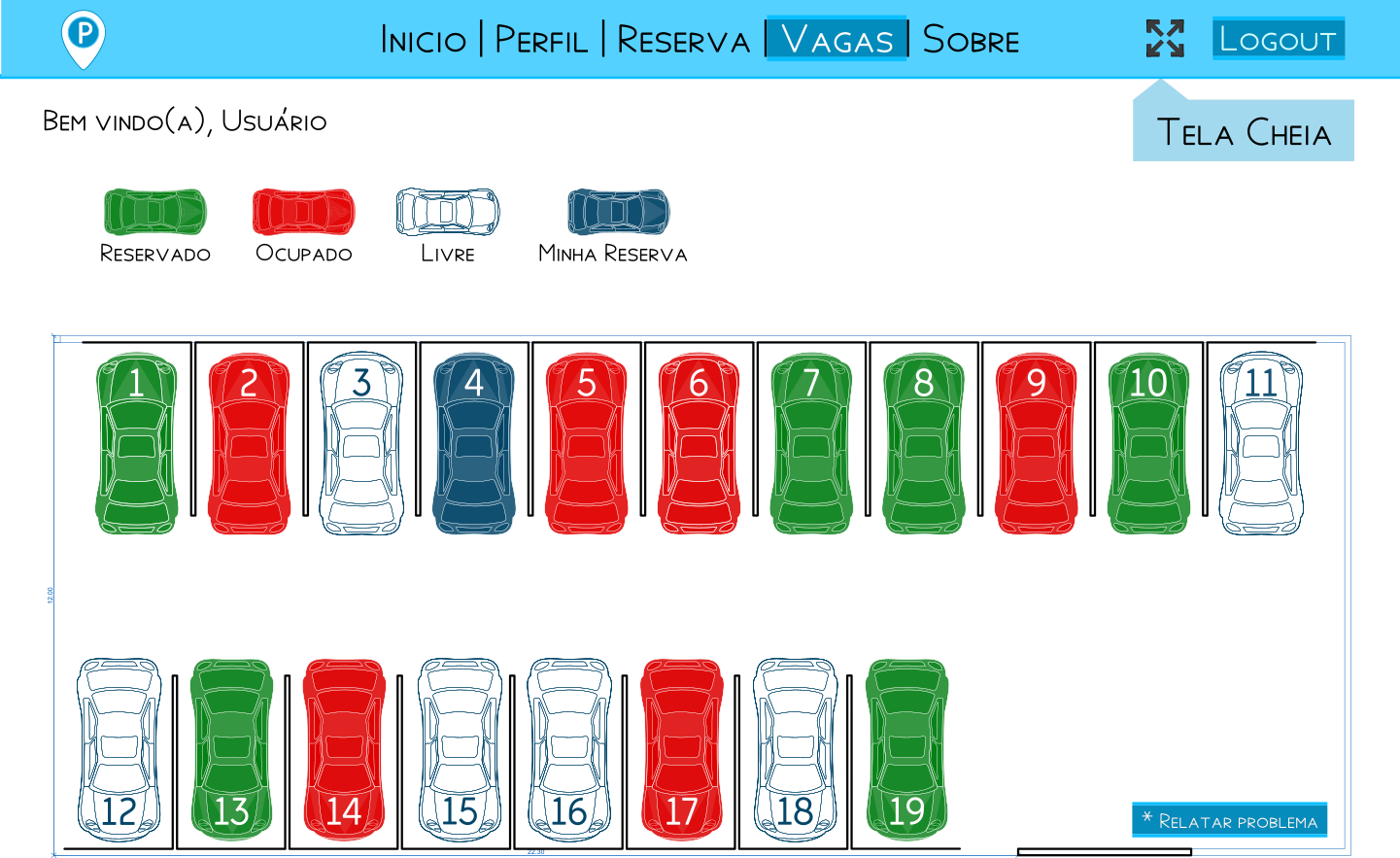
Esse tópico apresenta os protótipos de tela do software definindo a interface para supor o funcionamento do sistema.

1. – Tela de *Login*



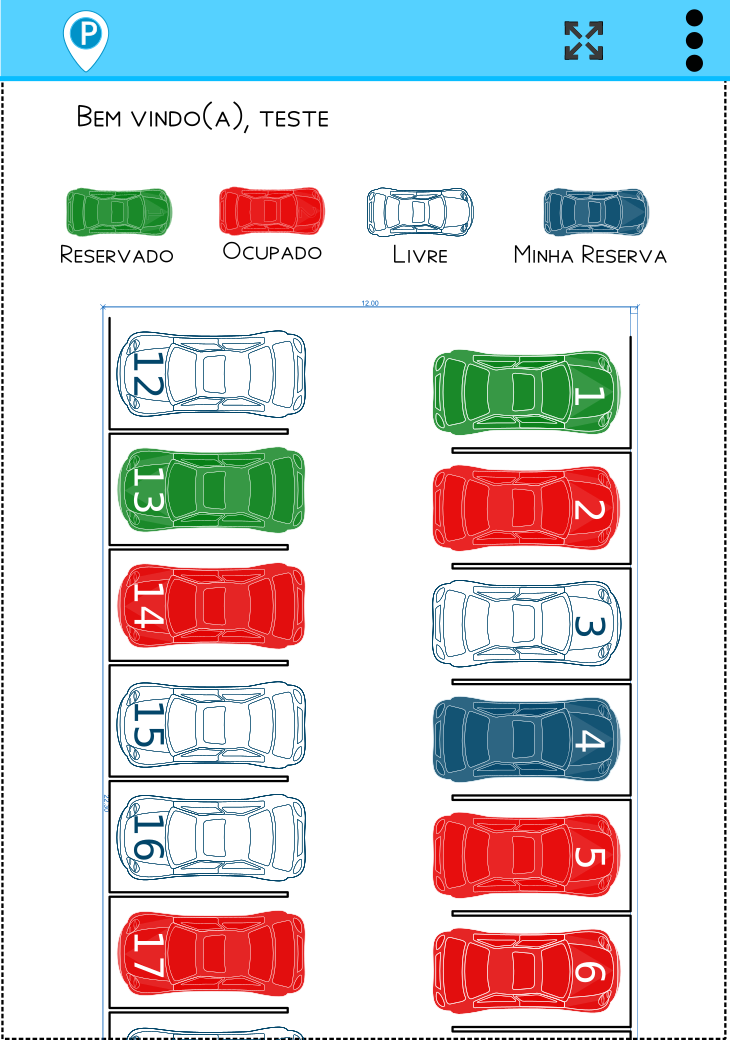
Fonte: Autores

1. – Tela de vagas para *Desktop*



Fonte: Autores

1. – Tela de vagas para Celulares



Fonte: Autores

APENDICE D – Modelo Canvas

APENDICE E – Resultados finais