

**INSTITUTO  
FEDERAL**

Goiás

Instituto Federal de Goiás

Câmpus Formosa

Tecnologia em Análise e Desenvolvimento de Sistemas

<http://www.ifg.edu.br/formosa>

**SISTEMA *WEB* PARA CONTROLE SEGURO DO ACERVO BIBLIOGRÁFICO DO  
IFG - CÂMPUS FORMOSA UTILIZANDO ARDUINO E RFID**

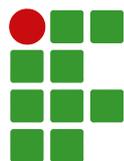
***PEDRO COSTA DIAS***

**Trabalho de Conclusão de Curso**

FORMOSA

2021





**INSTITUTO  
FEDERAL**

Goiás

Instituto Federal de Goiás

Câmpus Formosa

Tecnologia em Análise e Desenvolvimento de Sistemas

<http://www.ifg.edu.br/formosa>

**SISTEMA *WEB* PARA CONTROLE SEGURO DO ACERVO  
BIBLIOGRÁFICO DO IFG - CÂMPUS FORMOSA UTILIZANDO  
ARDUINO E RFID**

Pedro Costa Dias

*Trabalho de Conclusão de Curso apresentado ao Departamento de Áreas Acadêmicas da Instituto Federal de Goiás campus Formosa, como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.*

Orientador: Prof. M.e Mario Teixeira Lemes

FORMOSA

2021

---

Pedro Costa Dias

Sistema *Web* para Controle Seguro do Acervo Bibliográfico do IFG - Câmpus Formosa utilizando Arduino e RFID/ Pedro Costa Dias. – FORMOSA, 2021-72 p.; 30 cm.

Orientador Prof. M.e Mario Teixeira Lemes

Trabalho de Conclusão de Curso – Instituto Federal de Goiás, 2021.

1. IFG 2. Desenvolvimento de software 3. Aplicação *Web* 4. IoT I. Orientador: Prof. M.e Mario Teixeira Lemes. II. Instituto Federal de Goiás. IV. Título: Sistema *Web* para Controle Seguro do Acervo Bibliográfico do IFG - Câmpus Formosa utilizando Arduino e RFID

CDU 02:141:005.7

---

## ATA DA SESSÃO DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

Aos **dezesete dias do mês de março do ano de 2021**, às **14h00**, reuniram-se **virtualmente** no Instituto Federal de Goiás os componentes da Banca Examinadora prof. M.e **Mario Teixeira Lemes** (orientador), Prof<sup>a</sup>. M.<sup>a</sup> **Iasmini Virgínia Oliveira Lima** e prof. M.e **Sirlon Thiago Diniz Lacerda**, sob a presidência do primeiro, para procederem à avaliação da defesa do Trabalho de Conclusão de Curso “**Sistema Web para Controle Seguro do Acervo Bibliográfico do IFG – Câmpus Formosa Utilizando Arduino e RFID**”, de autoria de **Pedro Costa Dias**, discente do **Curso Superior em Tecnologia e Análise de Sistemas do IFG/Formosa**. Após a arguição dos membros da banca, chegou-se à conclusão que o trabalho foi \_\_\_\_\_ com nota \_\_\_\_, considerando-se integralmente cumprido este requisito para fins de obtenção de título de tecnólogo em análise e desenvolvimento de sistemas pelo Instituto Federal de Goiás.

Termo de aprovação do TCC (Resolução/IFG nº 28, de 11 de Agosto de 2014)

- Aprovação sem restrições (Art. 23. §2º)
- Aprovação com indicação de correções (Art. 23. §3º)
- Não-aprovação

---

Prof. M.e Mario Teixeira Lemes

---

Prof.<sup>a</sup> M.<sup>a</sup> Iasmini Virgínia Oliveira Lima

---

Prof. M.e Sirlon Thiago Diniz Lacerda

**TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO  
NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG**

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

**Identificação da Produção Técnico-Científica**

- |  |   |
|--|---|
| <input type="checkbox"/> Tese  | <input type="checkbox"/> Artigo Científico              |
| <input type="checkbox"/> Dissertação                                 | <input type="checkbox"/> Capítulo de Livro              |
| <input type="checkbox"/> Monografia – Especialização                 | <input type="checkbox"/> Livro                          |
| <input type="checkbox"/> TCC - Graduação                             | <input type="checkbox"/> Trabalho Apresentado em Evento |
| <input type="checkbox"/> Produto Técnico e Educacional - Tipo: _____ |   |

Nome Completo do Autor:

Matrícula:

Título do Trabalho:

**Restrições de Acesso ao Documento**

Documento confidencial:  Não  Sim, justifique: \_\_\_\_\_

Informe a data que poderá ser disponibilizado no ReDi/IFG: \_\_\_/\_\_\_/\_\_\_

O documento está sujeito a registro de patente?  Sim  Não

O documento pode vir a ser publicado como livro?  Sim  Não

**DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA**

O/A referido/a autor/a declara que:

- i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- ii. obteve autorização de quaisquer materiais incluídos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

\_\_\_\_\_, \_\_\_\_/\_\_\_\_/\_\_\_\_.  
Local Data

\_\_\_\_\_  
Assinatura do Autor e/ou Detentor dos Direitos Autorais

*Dedico este trabalho primeiramente ao Senhor meu Deus,  
meu guia e protetor que me acompanha no caminho  
percorrido nesta Terra e nas próximas. Agradeço também à  
minha mãe Valdomira que tenho como modelo de pessoa a  
ser seguida e meu pai Ronaldo que me motivou e serviu de  
inspiração durante todo trabalho.*

# Agradecimentos

Agradeço à minha mãe que sempre me apoiou de todas as formas possíveis na minha vida e que sempre insistiu que deveria me esforçar a estudar pois se o trabalho edifica o homem, então o conhecimento e os estudos o preenchem para torná-lo um homem de valor.

Agradeço também ao meu pai que em nenhum momento não esteve lá para dar suporte no eu precisasse, sendo desde bens materiais até conselhos e inspirações que tenho certeza que vou carregar por toda minha vida.

Agradeço aos meus queridos amigos que fiz durante o curso, entre eles Gustavo, Mário e Stefany, pessoas incríveis que com toda certeza mudaram minha vida e a forma que vejo o mundo, e que apesar de não poder citar todos, sempre lembrarei do companharismo e dos momentos compartilhados com cada um deles.

Agradeço ao todo o corpo docente do IFG - Câmpus Formosa, responsável por instigar minha fome por conhecimento e fazer com que buscasse aperfeiçoamento.

Por fim, agradeço ao meu querido orientador Mario, que desde a primeira matéria na qual foi meu professor, acreditou, instruiu e me orientou, com conselhos e broncas que me fizeram chegar até aqui.

A todas essas pessoas, o mais sincero *OBRIGADO!*

*"Bem-aventurado o homem que acha sabedoria, e o homem que adquire conhecimento, porque é melhor a sua mercadoria do que artigos de prata, e maior o seu lucro que o ouro mais fino, mais preciosa do que os rubis, e tudo o que mais possas desejar não se pode comparar a ela"*

*—PROVÉRBIOS 3:13-15.*



# Resumo

Avanços tecnológicos na minituarização dos dispositivos e nas tecnologias de comunicação sem fio permitem a expansão de serviços relacionados à Internet das Coisas (IoT) e aos objetos inteligentes. IoT é um paradigma de computação que permite que objetos inteligentes tenham poder computacional e capacidade de transmissão da informação, podendo ser usado nos mais diversos cenários tais como voltados para saúde, agricultura, segurança, entre outros. Nesse contexto, o objetivo deste Trabalho de Conclusão de Curso (TCC) é especificar e desenvolver um sistema *Web* que, juntamente com o sistema embarcado IoT Arduino e a tecnologia de identificação por rádio frequência, permitam o controle seguro do acervo bibliográfico do IFG-Câmpus Formosa, trazendo maior segurança ao patrimônio literário da Instituição ao evitar esquecimentos de registro de saída (empréstimos) e/ou furto de materiais.

**Palavras-chave:** Sistema Web, Internet das Coisas, Identificação por Radio Frequência, Controle de Acervo Bibliográfico.



# Abstract

Technological advances in miniaturization of devices and wireless communication technologies allow the expansion of services related to the Internet of Things (IoT) and intelligent objects. IoT is a computing paradigm that allows intelligent objects to have computational power and the ability to transmit information, and can be used in the most diverse scenarios such as health, agriculture, security, among others. In this context, the objective of this Course Conclusion Work (TCC) is to specify and develop a Web system that, together with the embedded IoT Arduino system and radio frequency identification technology, allow the safe control of the bibliographic collection of IFG-Câmpus Formosa, bringing greater security to the Institution's literary heritage by avoiding forgetfulness of check-out (loans) and / or theft of materials.

**Keywords:** Web System, Internet of Things, Radio Frequency Identification, Control of Bibliographic Collection.



# Lista de Figuras

2.1	Funcionamento geral de um SGBD. Fonte: <a href="https://www.oficinadanet.com.br/post/16631-o-que-e-um-sgbd">https://www.oficinadanet.com.br/post/16631-o-que-e-um-sgbd</a> . . . . .	27
4.1	Visão Geral do Sistema . . . . .	36
4.2	Diagrama de Casos de Uso . . . . .	37
4.3	Diagrama de Implementação . . . . .	38
4.4	Diagrama de Sequência . . . . .	40
4.5	Arquitetura do Banco de Dados . . . . .	41
4.6	Protótipo do Sistema Arduino . . . . .	41
4.7	Fluxograma do Sistema <i>Web</i> . . . . .	43
4.8	Página inicial do Sistema <i>Web</i> . . . . .	44
4.9	Aba de Listagem de <i>tags</i> . . . . .	45
4.10	Mensagem de confirmação para leitura de <i>tag</i> . . . . .	46
4.11	Aba de listagem de <i>tag</i> após a leitura da <i>tag</i> . . . . .	47
4.12	Mensagem de confirmação de aluguel . . . . .	47
4.13	Aba de listagem de <i>tag</i> quando o livro lido já está alugado . . . . .	48
4.14	Mensagem de confirmação de devolução de livro . . . . .	49
4.15	Aba de listagem de <i>tag</i> quando o livro não está cadastrado . . . . .	50
4.16	Aba de cadastro do livro . . . . .	50
4.17	Mensagem de confirmação do cadastro de material bibliográfico . . . . .	51
4.18	Aba de listagem de <i>tag</i> após o processo de leitura . . . . .	52



# Lista de Acrônimos

**IoT - Internet of Things.**

**RFID - Radio Frequency Identification.**

**IFG - Instituto Federal de Educação, Ciência e Tecnologia de Goiás.**

**TCC - Trabalho de Conclusão de Curso.**

**HD - Hard Disk.**

**SSD - Solid State Drivers.**

**SGBD - Sistema Gerenciador de Banco de Dados.**

**SQL - Structured Query Language.**

**DDL - Data Definition Language.**

**DML - Data Manipulation Language.**

**IDE - Integrated Development Environment.**

**PHP - Hypertext Preprocessor.**

**HTML - Hyper Text Markup Language.**

**UML - Unified Modeling Language.**

**UP - Unified Process.**

**WiFi - Wireless Fidelity.**

**BD - Banco de Dados.**



# Sumário

<b>1</b>	<b>Introdução</b>	<b>19</b>
<b>2</b>	<b>Referencial Teórico</b>	<b>25</b>
2.1	Introdução . . . . .	25
2.2	Banco de Dados . . . . .	25
2.2.1	Modelo Relacional . . . . .	26
2.2.2	SQL . . . . .	26
2.2.3	SGBD . . . . .	26
2.3	Microcontroladores . . . . .	27
2.3.1	Linguagem C . . . . .	27
2.3.2	Arduino . . . . .	28
2.3.3	RFID . . . . .	28
2.4	Sistema <i>Web</i> . . . . .	28
2.4.1	Lógica Computacional e Linguagem de Programação . . . . .	28
2.4.2	PHP . . . . .	29
2.4.3	IDE . . . . .	29
2.4.4	XAMPP . . . . .	29
2.5	Arquitetura de <i>Software</i> . . . . .	29
2.5.1	UML . . . . .	30
2.5.2	Processo Unificado . . . . .	30
2.6	Considerações Finais . . . . .	30
<b>3</b>	<b>Material e Métodos</b>	<b>31</b>
3.1	Introdução . . . . .	31
3.2	Metodologia da Pesquisa . . . . .	31
3.3	Ferramentas Utilizadas . . . . .	32
3.4	Considerações Finais . . . . .	33

<b>4</b>	<b>Resultados</b>	<b>35</b>
4.1	Introdução . . . . .	35
4.2	Visão Geral do Sistema . . . . .	35
4.2.1	Casos de Uso . . . . .	37
4.2.2	Diagrama de Implementação . . . . .	38
4.2.3	Modo de Funcionamento . . . . .	38
4.3	Sistema de leitura de <i>tags</i> RFID com Arduino . . . . .	41
4.3.1	Descrição dos Componentes de <i>Hardware</i> . . . . .	42
4.3.2	Descrição do Funcionamento do Código para Execução no Arduino . .	42
4.4	Página <i>Web</i> . . . . .	43
4.4.1	Buscar por Livros . . . . .	44
4.4.2	Alugar Livros . . . . .	46
4.4.3	Devolver Livros . . . . .	48
4.4.4	Cadastrar Livros . . . . .	49
4.4.5	Remover Livros . . . . .	51
4.5	Considerações Finais . . . . .	52
<b>5</b>	<b>Conclusão</b>	<b>53</b>
	<b>Referências</b>	<b>55</b>
	<b>Apêndice</b>	<b>57</b>
.1	Documento de Requisitos . . . . .	59

# 1

## Introdução

De acordo com SINGER (2012), IoT (*Internet of Things*) ou "Internet das Coisas" pode ser caracterizado como objetos conectados em rede que processem e/ou produzam informação em tempo real e de forma autônoma. O termo IoT também aparece como sinônimo de ambientes conectados, cidades inteligentes, internet do futuro entre outros.

IoT é o modo como os objetos físicos estão conectados e se comunicando entre si e com o usuário, através de sensores inteligentes e softwares que transmitem dados para uma rede. Qualquer objeto pode se tornar inteligente. Alguns exemplos de objetos inteligentes: um relógio ou uma geladeira, carros, máquinas, computadores e smartphones (CASTRO, 2019).

CARRION (2019) expressa em seu trabalho que o mundo computacional e o acesso a internet tem interação em sua maior parte com uma tela. Porém, iniciativas e previsões de um mundo de objetos conectados para além da tela de um computador existem desde antes da primeira geração de computadores. A autora também refere-se a IoT como objetos, ambientes, veículos e roupas inteligentes com capacidades sensoriais que estão capacitados terão cada vez mais informações associadas a eles, podendo se conectar e se comunicar uns com os outros e com demais dispositivos habilitados para a *Web*.

IoT emergiu dos avanços de várias áreas como sistemas embarcados, microeletrônica, comunicação e sensoriamento e tem recebido bastante atenção tanto da academia quanto da indústria, devido ao seu potencial de uso nas mais diversas áreas das atividades humanas. IoT é considerada a extensão da Internet, que proporciona aos objetos do dia a dia capacidade

computacional e de comunicação permitindo assim se conectarem à Internet (SANTOS, 2016).

Segundo CARVALHO (2016), a implementação de aplicações IoT em bibliotecas se torna oportuna, especialmente quando o serviço está relacionado ao atendimento ao público, envolvendo quesitos como segurança, facilidade de acesso, reserva e agendamento de livros, entre vários outros pontos que poderiam ser de melhoria com implementação de simples dispositivos tais como leitores de código de barras até *chips* de Identificação de Rádio Frequência (RFID - *Radio Frequency Identification*).

ZURITA (2011) defende que a popularização dos microprocessadores na década de 1980 foi responsável pelo barateamento da tecnologia. Logo, os processadores que eram totalmente restritos a computadores corporativos e algumas grandes máquinas, podem ser encontrados praticamente por toda a parte. Automóveis, televisores, brinquedos, máquinas de lavar, telefones, aparelhos de som, relógios de pulso, cartões de banco e até mesmo etiquetas de identificação descartáveis. A adoção de microprocessadores fora do escopo dos computadores pessoais e corporativos impulsionou os sistemas computacionais embarcados.

Sistemas embarcados estão presentes em quase todas as atividades humanas, e devido ao custo de produção tecnológica ficando cada vez menores, a tendência é que esse tipo de sistema seja cada vez mais utilizado e aumente sua presença no mercado e no cotidiano das pessoas. O projeto desse tipo de sistema é extremamente complexo por envolver conceitos e questões pouco discutidas pela computação, que toma como propósito maior a computação de propósitos gerais (CARRO, 2003).

Um sistema embarcado é um sistema, equipamento ou circuito integrado que tem a capacidade computacional dentro dele, completo e independente, porém feito para realizar apenas a tarefa determinada à ele. Não há espaço para flexibilidade, o mais perto disso seria na implementação de funcionalidades em novas versões e *upgrades* de sistema (CUNHA, 2007). Segundo TAVARES (2018), placas embarcadas foram desenvolvidas para serem dispositivos “invisíveis” e difundir-se no nosso cotidiano e para dedicarem-se a apenas um processo de maneira contínua.

Em 2005, no Instituto de Interatividade e *Design* na cidade de Ivrea (Itália), foi criado um microcontrolador intitulado de Arduino pelos professores de computação física Massimo Banzi

---

e David Cuartielles. Esse microcontrolador vem sendo muito utilizado para o aprendizado de eletrônica e programação, podendo ser utilizado em projetos de automação residencial e robótica. O *hardware* é livre e a linguagem de programação simples com a possibilidade de acoplamento de diversos sensores, aumentando assim suas funcionalidades (SILVA, 2015).

BANZI (2011) diz em seu livro que a placa Arduino pode ser utilizada para desenvolvimento de objetos interativos independentes, ou conectada a *softwares* de computadores. A simplicidade de manuseio e a alta acessibilidade são fatores que propulsionam a integração de dispositivos embarcados e *softwares* diversos.

De acordo com OLIVEIRA (2006), RFID é uma tecnologia que utiliza ondas eletromagnéticas para identificar objetos, pessoas e/ou animais. Sistema RFID é composto basicamente por três componentes: o transpônder que carrega a *tag* de identificação, a antena ou algum dispositivo de leitura que é responsável pela emissão do campo eletromagnético que faz com que o transpônder responda, e o computador que é conectado ao dispositivo de leitura e recebe os dados provenientes do transpônder.

A tecnologia RFID tem uma vasta abrangência de possibilidades de atuação, podendo ser usadas nas mais diversas situações, tais como: controle de estoque, monitoramento de animais, busca de produtos, sistema contra roubos, sistema de pedágio sem necessidade de parada em rodovias, monitoramento de pacientes, controle de correspondências, controle de acesso, identificação de bagagens em aeroportos, etc (GONÇALVES, 2008).

De acordo com PINHEIRO (2017), RFID oferece inúmeras possibilidades de aplicação, apresentando soluções para os sistemas de rastreamento e identificação com diversas vantagens. RFID é um tipo de tecnologia de identificação automática, uma vez que objetos podem ser etiquetados e identificados de forma estruturada, um *microchip* pode fornecer automaticamente informações sobre um produto, tais como localização física, código do fabricante, fornecedor, comprador, etc.

ROCHA (2019) apresentou em sua monografia um sistema de controle de acesso para um laboratório, utilizando RFID's e uma placa embarcada ESP32-DevKitC que realiza a comunicação com o servidor e banco de dados na Internet. Tal sistema também é capaz de realizar o monitoramento do fluxo de entrada e saída de pessoas, além de fornecer relatórios

relacionadas às questões de acesso.

COSTA (2018) propôs em seu trabalho de conclusão de curso um sistema de gerência de estoque que possibilitava a realização de controle de acesso, contagem e identificação de mercadoria através do uso da tecnologia RFID. O sistema é composto por duas partes, um sistema *Web* e um protótipo de *hardware* que permite as funções de leitura do RFID e de comunicação com a Internet.

SOARES (2018) criou um sistema similar aos supracitados, utilizando o microcontrolador Arduino uno juntamente com a linguagem de programação Delphi. Neste caso, o objetivo foi realizar o controle e verificação da presença dos alunos com o auxílio de *chips* RFID's para identificação e obtenção do registro de frequência dos alunos.

NASCIMENTO (2016) propõe um uso para a tecnologia RFID ainda mais complexo e desafiador. Além de servir para identificação de objetos, pessoas e animais, tal sistema seria utilizado para localização automática de objetos através de medidas de força de sinal. Em sua dissertação de mestrado, o autor apresenta algoritmos de localização para sistemas RFID e realiza a análise da precisão desses algoritmos usando como cenário de configuração a localização de livros com etiquetas passivas (que não tem bateria) em bibliotecas.

Já os autores em FREITAS (2020) apresentam um sistema de registro de entrada e saída de itens patrimoniados no almoxarifado, fazendo uso de artefatos como o microcontrolador ESP8266, bibliotecas da linguagem de programação C, e banco de dados gratuito MySQL.

Sabendo que a tecnologia RFID pode ser utilizada em conjunto com a placa embarcada Arduino, no contexto das mais diversas aplicações IoT, presume-se que seja possível o desenvolvimento de um sistema computacional de controle de livros para a biblioteca do IFG-Câmpus Formosa. O controle e a segurança dos livros da biblioteca gera proteção do patrimônio e sua utilização permitirá maior controle, por parte dos bibliotecários, e manutenção do patrimônio literário ao evitar possíveis esquecimentos e/ou furtos.

## **Objetivos**

### **Objetivo Geral**

Desenvolver um sistema de controle seguro do acervo bibliográfico para o uso dos funcionários da biblioteca do IFG - Câmpus Formosa utilizando a tecnologia RFID e o microcontrolador Arduino, gerenciável por meio de uma aplicação *Web* que permita controle de cadastro de novos materiais bibliográficos, edição da situação atual (alugado ou não alugado) e a remoção (efetivação do empréstimo) com base na leitura das *tags* de radio frequência presentes nos materiais bibliográficos.

### **Objetivos Específicos**

1. Realizar análise e especificação dos requisitos para o desenvolvimento das funcionalidades que devem ser desenvolvidas no Arduino e na aplicação *Web* proposta.
2. Desenvolver o sistema de leitura para salvamento de dados no banco através do Arduino e o sensor RFID.
3. Desenvolver um sistema *Web* que permita o controle seguro de cadastro, remoção e edição da situação (alugado/não-alugado) de um material bibliográfico .

### **Descrição dos Capítulos**

Este TCC está dividido em 5 (cinco) capítulos. Neste Capítulo foi apresentado o estado da arte de aplicações IoT, a justificativa para realização desta pesquisa bem como os objetivos geral e específicos.

No Capítulo 2 apresentamos um Referencial Teórico, com objetivo de apresentar uma descrição teórica dos principais artefatos que foram utilizados como base para o desenvolvimento deste trabalho.

Já no Capítulo 3 expomos a metodologia da pesquisa utilizada para satisfação dos objetivos definidos, compreendendo as etapas de análise/elucidação dos requisitos de sistema e processo de desenvolvimento. Apresentamos também os materiais e métodos utilizados na

construção dos sistemas propostos.

O Capítulo 4 fornece os resultados obtidos bem como a discussão do sistema para controle seguro do acervo bibliográfico aplicado ao IFG-Câmpus Formosa. Finalmente, no Capítulo 5 são abordadas as contribuições deste trabalho acadêmico e a descrição de possíveis trabalhos futuros.

# 2

## Referencial Teórico

### 2.1 Introdução

Neste capítulo é realizado uma breve explicação sobre artefatos utilizados nesse trabalho, tendo como objetivo a contextualização de leitores menos familiarizados com o tema. É apresentado também ideias gerais sobre os principais conceitos e componentes sobre os quais esse trabalho foi construído.

### 2.2 Banco de Dados

"Bancos de dados são coleções de dados interligados entre si e organizados para fornecer informações". (ALVES, 2013). Este é um conceito amplamente utilizado em computação, sempre que houver uma informação e essa mesma precisar ser armazenada, um banco de dados é criado para realizar essa função.

Para criação de um banco de dados que possa ser utilizado em um projeto são necessárias um dispositivo de *hardware* para armazenamento, HD (*Hard Disk*) ou SSD (*Solid State Disk*), o SGBD (Sistema Gerenciador de Banco de Dados) é uma linguagem de consulta para manipulação das informações presentes no banco de dados.

### 2.2.1 Modelo Relacional

O modelo relacional foi proposto em 1970 por Edgar Codd, permitindo representação simples dos dados e trazendo facilidade para realizar consultas consideradas complexas (BALDO, 2005). A modelagem relacional vê os dados como um conjunto de tabelas que podem se relacionar entre si, compostas de 'linhas' (registros), onde cada registro armazena um valor ou um conjunto de informações relacionadas separadas por colunas que dividem e classificam cada setor dessas 'linhas'.

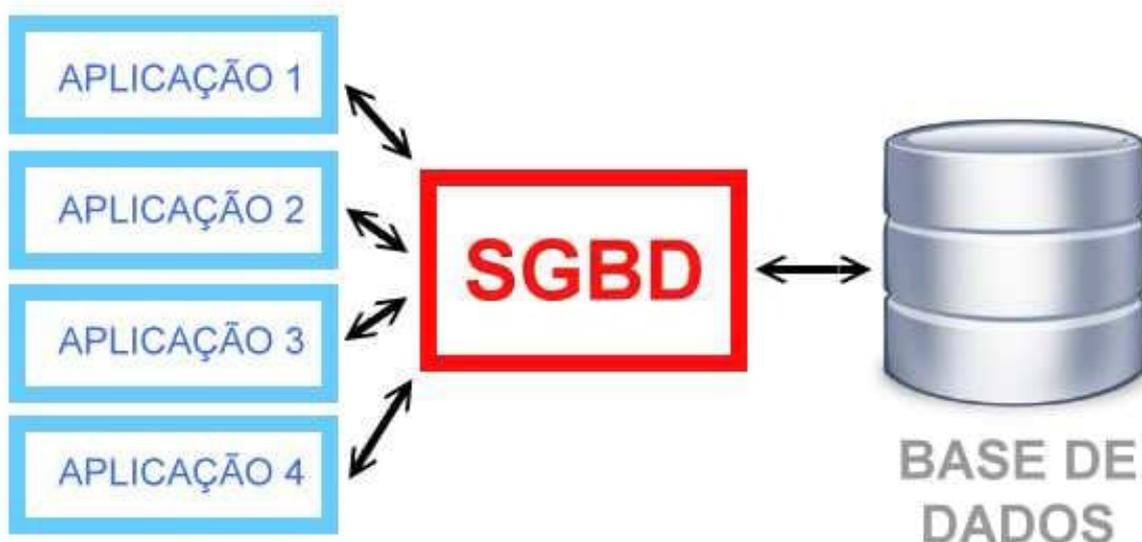
### 2.2.2 SQL

Segundo OLIVEIRA (2013), SQL (*Structured Query Language*) é a linguagem padrão universal para manipular bancos de dados relacionais através dos SGBDs, utilizada para interagir com o SGBD e executar várias tarefas como inserir e alterar registros, criar objetos, gerenciar usuários, consultar informações, controlar transações, dentre outras. Todas as operações realizadas no banco de dados podem ser solicitadas ao SGBD utilizando esta linguagem.

SQL é dividida em DDL (*Data Definition Language* ou Linguagem de Definição de Dados) e a DML (*Data Manipulation Language* ou Linguagem de Manipulação de Dados), onde a DDL é utilizada para manipular a arquitetura do banco de dados, ou seja, como os dados serão armazenados, e a DML para manipulação da informação.

### 2.2.3 SGBD

De acordo com (OLIVEIRA, 2013), SGBD é responsável pelo gerenciamento do banco e por diversas ações, tais como salvar os dados no dispositivo de armazenamento, manter em memória os dados mais acessados, ligar dados e metadados, criptografar informações, disponibilizar interface para programas e usuários externos, controlar o acesso as informações e manter cópias (*backup*) dos dados para recuperação em casos de falha e garantir as transações em um banco de dados. A Figura 2.1 mostra o funcionamento geral de um SGBD.



**Figura 2.1:** Funcionamento geral de um SGBD. Fonte: <https://www.oficinadanet.com.br/post/16631-o-que-e-um-sgbd>

## 2.3 Microcontroladores

De acordo com ARAUJO (2020), um microcontrolador é um computador muito pequeno capaz de realizar determinadas tarefas de maneira eficaz e sob um tamanho altamente compacto. A importância dos microcontroladores é devido ao fato da alta demanda por sistemas embarcados no mercado e por serem sistemas computacionais compactos e de custo acessível que atendem a uma necessidade específica. A linha Atmel é utilizada nas placas Arduino, sendo bastante difundida devido à sua simplicidade e acessibilidade.

### 2.3.1 Linguagem C

Diversos microcontroladores disponíveis no mercado contam com compiladores de linguagem C para o desenvolvimento de *software*. Um aspecto favorável da utilização da linguagem C é a eficiência. Por se tratar de uma linguagem de baixo nível, a proximidade a linguagem de máquina e ao *hardware* faz de C uma linguagem extremamente rápida e que utiliza com eficiência os recursos computacionais disponíveis. (PEREIRA, 2007).

### 2.3.2 Arduino

O Arduino é uma plataforma formada por dois componentes: a placa, que é o *hardware* que pode ser utilizado para a construção dos projetos e a IDE, que é o *software* no qual escrevemos as instruções que devem ser executadas (MOTA, 2015).

O Arduino permite que componentes adicionais de *hardware* (sensores e atuadores) sejam conectados, possibilitando uma variedade de combinações, expandido os serviços genuinamente ofertados por este sistema embarcado.

### 2.3.3 RFID

Segundo PETENATE (2019), RFID trata-se de uma evolução do código de barras e que permite um método de identificação automática por sinais de rádio. A comunicação é realizada por meio de troca de informações entre etiquetas RFID e a base transmissora, onde um sinal de radio é enviado de uma base transmissora a um *microchip* que possui informações previamente gravadas.

## 2.4 Sistema Web

Sistema *Web* é um *software* hospedado na Internet acessível por meio de um navegador. A vantagem de aplicações *Web* é sua interoperabilidade, pois a mesma pode ser acessada praticamente de qualquer lugar do mundo, desde que exista conexão à Internet (BÔAS, 2017).

### 2.4.1 Lógica Computacional e Linguagem de Programação

A lógica de programação é um conceito fundamental no processo de desenvolvimento de *software* pois detalha os processos que devem ser executadas por aquele programa de computador.

Uma linguagem de programação é utilizada para que o computador execute as instruções. Uma comparação simples pode ser realizada: a lógica é a mensagem que você quer passar ao sistema, e as linguagens de programação são o idioma. Existe uma grande variedade de linguagens de programação disponíveis e sua utilização vai depender do interesse e da necessidade do desenvolver de *software*.

### 2.4.2 PHP

De acordo com CONVERSE (2003), PHP é uma linguagem de programação utilizada no contexto de desenvolvimento de sistemas *Web*. Uma vantagem é a facilidade com a qual ela permite o desenvolvedor construir funcionalidades. Outra característica interessante desta linguagem é a possibilidade da integração com a linguagem HTML (Linguagem de Marcação de Hipertexto).

### 2.4.3 IDE

De acordo com ANDRADE (2020), IDE (*Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado) é um *software* que auxilia no desenvolvimento de aplicações. O objetivo é facilitar diversos processos ligados ao desenvolvimento, combinando ferramentas úteis em uma única interface gráfica de usuário.

As IDEs auxiliam em todo o processo de desenvolvimento de uma aplicação, provendo diversos benefícios, tais como a análise de código escrito (importante na identificação de defeitos de *software*), ferramentas que auto completam trechos de códigos, dentre outras funcionalidades.

### 2.4.4 XAMPP

O XAMPP é um pacote com os principais servidores de código aberto do mercado, incluindo banco de dados MySQL e Apache com suporte para a linguagem PHP. É possível rodar sistemas *Web* de forma local, ou seja, sem a necessidade de hospedar seu projeto em um servidor, o que facilita e agiliza o desenvolvimento (HIGA, 2012).

## 2.5 Arquitetura de *Software*

É uma sub-área da Engenharia de *Software* que analisa os componentes necessários que um *software* deve satisfazer para funcionamento de acordo com os requisitos. Trata-se de um conceito abstrato que trata da relação entre o mapeamento dos componentes de um *software* e os detalhes que são considerados no processo de desenvolvimento (ARAUJO, 2020).

A arquitetura de *software* permite a análise/elucidação dos requisitos, a construção de um documento de requisitos do sistema bem como os diagramas de UML (*Unified Modeling*

*Language* ou Linguagem de Modelagem Unificada). Nesse sentido, entende-se que é o processo inicial que deve ser realizado para construção de qualquer aplicação computacional.

### 2.5.1 UML

Segundo VENTURA (2019), UML é uma linguagem de notação para ilustração dos processos de comunicação presentes em sistemas computacionais. Esta linguagem é expressa através de diagramas compostos de desenhos e formas geométricas. A UML é dividida em 2 (dois) grupos: diagramas estruturais e comportamentais, no qual o primeiro grupo trata a descrição da arquitetura e a forma que o sistema é construído, observando componentes do sistema como suas classes e métodos. Já os diagramas comportamentais descrevem como o sistema se comporta, ou seja, seu modo de funcionamento.

### 2.5.2 Processo Unificado

De acordo com VENTURA (2019), o Processo Unificado (*Unified Process* - UP) é uma forma de utilização dos modelos convencionais de desenvolvimento de forma que possam se adequar ao Desenvolvimento Ágil de *software*. UP foi criado em 1995 por Jim Rumbaugh, Grady Booch e Ivar Jacobson para suportar o paradigma de Orientação a Objetos.

## 2.6 Considerações Finais

Neste Capítulo foram apresentados conceitos inerentes ao contexto de IoT, tais como microcontroladores, dispositivos computacionais (Arduino) e tecnologias RFID. Foram abordados também conceitos clássicos de computação, tais como linguagens de programação, banco de dados e interfaces de desenvolvimento. Por fim, conceitos relacionados a arquitetura de *software* e a metodologia de desenvolvimento foram exemplificadas.

# 3

## Material e Métodos

### 3.1 Introdução

Este capítulo aborda a metodologia de pesquisa científica utilizada na realização deste trabalho bem como os materiais e métodos utilizados no escopo da solução apresentada no Capítulo 4, ou seja, a definição e o desenvolvimento de um sistema *Web* que, em conjunto com o sistema embarcado Arduino e a tecnologia RFID, forneçam gerenciamento seguro do material bibliográfico do IFG - Câmpus Formosa.

### 3.2 Metodologia da Pesquisa

Em relação ao método de pesquisa, este Trabalho de Conclusão de Curso (TCC) utilizou-se do método hipotético-dedutivo. Este método utiliza a dedução e o raciocínio lógico para, após as etapas do processo, chegar a uma conclusão verdadeira. A hipótese, apresentada no Capítulo 1, baseou-se na possibilidade do desenvolvimento de um sistema computacional de controle seguro do material bibliográfico do IFG-Câmpus Formosa, uma vez que é possível integrar a tecnologia RFID e o dispositivo embarcado Arduino.

Em relação a modalidade da pesquisa, a pesquisa bibliográfica foi utilizada para aprofundamento dos conhecimentos necessários para a realização da proposta deste TCC. Dessa forma, bases eletrônicas, como 'Google Acadêmico', 'SCOPUS', 'IEEE Xplore Digital Library', 'ACM Digital Library' e sites de busca foram utilizadas para construção do acervo bibliográfico

necessário para a realização da pesquisa. Foram considerados trabalhos publicados nos últimos 18 anos.

A metodologia de desenvolvimento baseada no UP foi utilizada para definição dos requisitos, construção do casos de uso, bem como no processo e análise de *software*. O UP mescla características das metodologias tradicionais e ágeis, que somada à diagramação UML permite a transformação de requisitos do sistema em *software*. UP também combina os ciclos de desenvolvimento iterativo e incremental, possibilitando ao desenvolvedor liberdade no processo de desenvolvimento.

### 3.3 Ferramentas Utilizadas

Para a ilustração e criação das figuras presentes no Capítulo 4 e dos diagramas da UML presentes no documento de requisitos do Apêndice .1, foi utilizado o *Lucid*<sup>1</sup>, um *software online* de diagramação e comunicação visual. Para prototipação do sistema físico do Arduino foi utilizada o *circuito.io*<sup>2</sup>, plataforma *online* que permite ao usuário realizar a prototipação de sistemas da placa Arduino uno.

Para o desenvolvimento do código do sistema Arduino foi utilizado a IDE Visual Studio na versão 16.8.31019.35 para o Sistema Operacional *Windows*. Já para a codificação do sistema *Web* foi utilizado a IDE *notepad++* na versão 7.9. Para permitir que a página *Web* fosse testada localmente utilizamos o pacote de servidores de código aberto XAMPP, em sua versão 3.2.2.

O armazenamento de dados foi realizado utilizando a plataforma *Clever Cloud*<sup>3</sup>. Esta plataforma integra o SGBD *phpMyAdmin* (versão 4.8.4) para o gerenciamento do banco de dados. Para modelagem do *front-end* foi utilizado o *Bootstrap*. Para desenvolvimento dos requisitos funcionais foi utilizado a linguagem de programação PHP na versão 7.2.34.

O *software* de versionamento de projetos *GitHub*<sup>4</sup> foi utilizado para o gerenciamento e controle dos códigos desenvolvidos. Todas as ferramentas supracitadas foram executadas sobre o sistema operacional *Windows 10*, em sua versão 1909.

---

<sup>1</sup><https://lucid.app/>

<sup>2</sup><https://www.circuito.io/>

<sup>3</sup><https://www.clever-cloud.com/>

<sup>4</sup><https://github.com>

### **3.4 Considerações Finais**

Neste Capítulo foram abordados os métodos e ferramentas utilizados no desenvolver deste trabalho, além de apresentarmos a classificação desta pesquisa científica. A pesquisa bibliográfica permitiu o aprofundamento em temas relacionados ao desenvolvimento deste trabalho, a citar: IoT, sistemas computacionais embarcados (Arduino), tecnologia RFID, dentre outras. O UP foi utilizado como base no processo de desenvolvimento, e com o auxílio dos diagramas da UML, o sistema de controle seguro para o material bibliográfico do IFG - Câmpus Formosa foi construído.



# 4

## Resultados

### 4.1 Introdução

Este Capítulo é responsável por abordar os módulos de composição do sistema *Web* de controle seguro do material bibliográfico do IFG - Câmpus Formosa. Neste momento, descrevemos a funcionalidade e as particularidades de cada um dos módulos, em termos de uso de dispositivos de *hardware* para a prototipação do sistema que envolve o Arduino, e os códigos utilizados para o funcionamento correto do sistema. Apresentamos o funcionamento geral do sistema proposto e, finalmente, realizamos uma discussão abordando suas principais funcionalidades.

### 4.2 Visão Geral do Sistema

O projeto foi dividido em 3 (três) partes, i) Sistema *Web* no qual o usuário pode interagir e fazer requisições, ii) Sistema Arduino que é responsável por dar suporte ao projeto ao fornecer tratamento relacionado as *tags* RFID e, iii) Banco de dados (BD) que, além de armazenar as informações pertinentes aos livros, atua como intermediador entre os 2 (dois) módulos de sistema supracitados.

O sistema *Web* e o sistema Arduino se comunicam através de tabelas auxiliares definidas no BD. A Figura 4.1 trás uma visão geral do sistema, com a definição dos principais componentes utilizados e seus inter-relacionamentos.

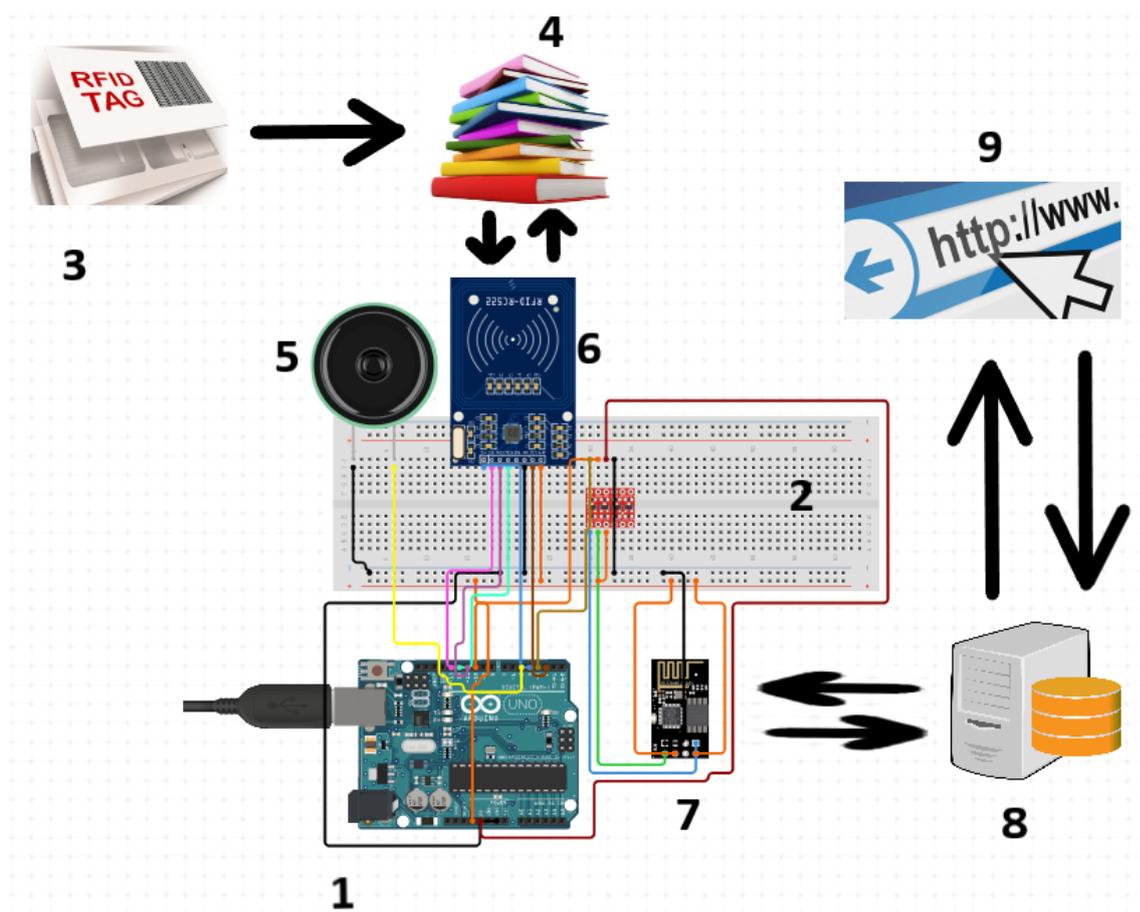


Figura 4.1: Visão Geral do Sistema

O sistema é composto pelos seguintes componentes de *hardware* e *software*:

1. Arduino Uno.
2. *Protoboard*.
3. *Tag* RFID.
4. Livros com *tags* RFID.
5. Auto-falante (*Buzzer*).
6. Módulo leitor de RFID.
7. Módulo *WiFi*.
8. Banco de dados (BD)
9. *Página Web*.

### 4.2.1 Casos de Uso

No diagrama de casos de uso apresentam-se 2 (dois) módulos do sistema, a citar: Sistema *Web* e Sistema *Arduino*. O diagrama elucida funcionalidades que cada dos módulos é capaz de executar. Para representação da página *Web* destacam-se as funções de cadastrar, excluir, alugar e devolver livros, além da função de solicitação de leitura de *tag*.

Na divisão do sistema *Arduino* destacam-se funções que chamam por outras funções, tais como 'Verificar situação do livro' que é invocada quando é identificado a saída de um livro. Caso a situação esteja como 'não-alugado', uma segunda função 'soar alarme' é acionada.

Tem-se também a função 'Verifica solicitações de leitura' que possui a funcionalidade de pesquisar no BD novas solicitações de leitura temporariamente realizadas pela página *Web*. No momento em que é identificado uma nova requisição, a função 'ler *tag* do livro' é invocada e realiza a leitura da *tag* para registro no BD. A Figura 4.2 mostra o diagrama de casos de uso em alto nível proposto para mapeamento das funcionalidades de cada um dos módulos.

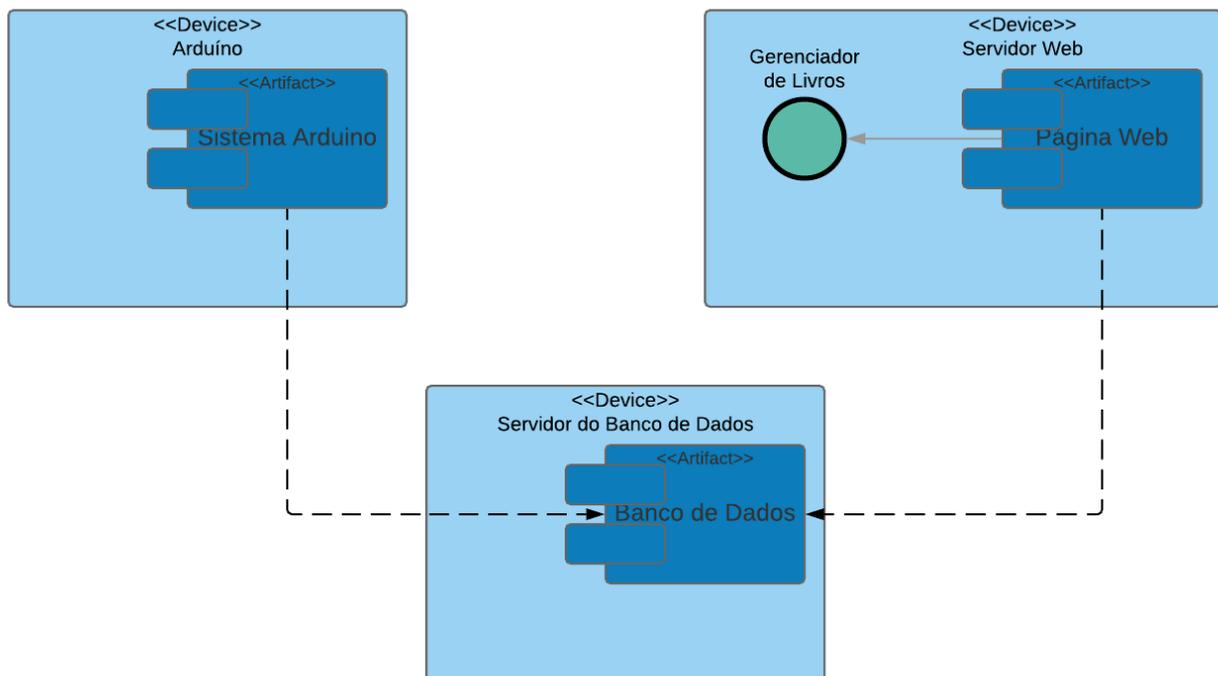


**Figura 4.2:** Diagrama de Casos de Uso

### 4.2.2 Diagrama de Implementação

O diagrama de implementação descreve a forma como os componentes do sistema estão divididos. Note na Figura 4.3 que os módulos página *Web* e Arduino convergem para o BD. Esse comportamento é justificado devido a dupla funcionalidade do BD: armazenamento das informações dos livros e componente intermediador da comunicação entre as dois módulos supracitados, atuando como repositório para compartilhamento de informações.

O Arduino e a Página *Web* dependem do BD para realização de suas tarefas, onde a página *Web* necessita do BD para realização das suas requisições e o arduino faz uso para buscar pelas mesmas.



**Figura 4.3:** Diagrama de Implementação

### 4.2.3 Modo de Funcionamento

O processo inicia-se a partir do momento em que algum usuário da biblioteca decide alugar um livro do acervo bibliográfico. Nesse momento, o visitante deve se direcionar a recepção para que o bibliotecário possa realizar o procedimento de empréstimo de material, através da página *Web* que permite a modificação da situação (alugado/não alugado) do livro no BD.

O BD vai armazenar informações relacionados as *tags*, títulos e situação dos livros.

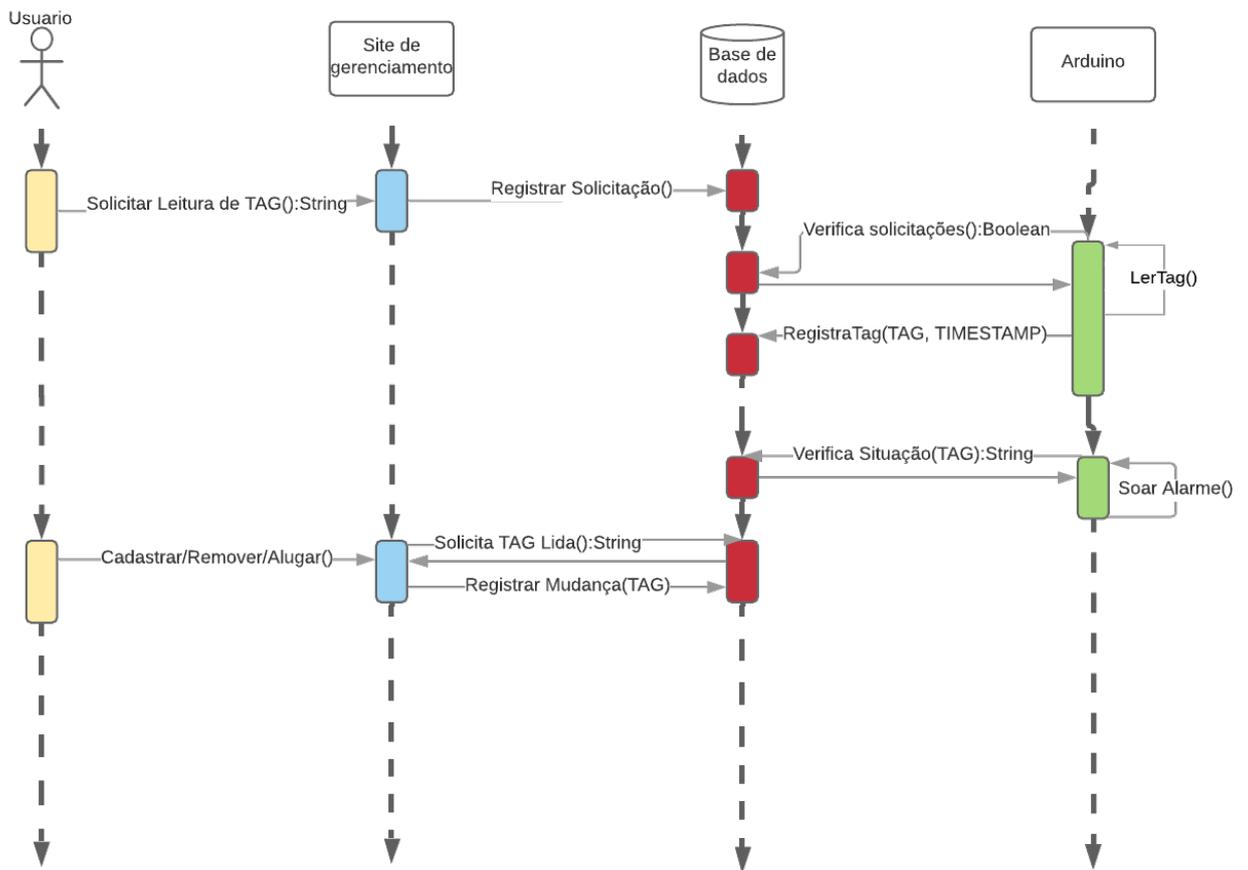
O campo situação dos livros no BD pode assumir 2 (dois) valores distintos: alugado ou não-alugado. O armazenamento dessa informação é necessário para que seja possível a identificação de possíveis esquecimentos e/ou furtos de material bibliográfico sem devido registro de saída.

Através do site de gerenciamento, além de modificar a situação de aluguel do livro, o bibliotecário é capaz de adicionar novos livros ou removê-los. Somente pela interface *Web* é possível fazer qualquer tipo de modificação na tabela de livros no BD. O Arduino tem permissão de escrita somente em tabelas auxiliares. Nas tabelas principais, ou seja, aquelas que armazenam as informações referentes aos títulos, o acesso do Arduino é apenas de leitura para possibilitar a realização de possíveis consultas.

O Arduino atua como intermediador, isto é, é através dele que os módulos funcionam em conjunto, fornecendo o processo de leitura de *tags*, comunicação com o BD e para ativação do sistema de alarme. A tabela 'solicitacao' é utilizada para informar ao Arduino que o sistema *Web* deseja realizar uma leitura de *tag*.

Quando a leitura é realizada pelo Arduino, a tabela auxiliar 'solicitacao' é consultada para verificar a possibilidade dessa leitura ser informada à página *Web*. Caso verdadeiro, o Arduino registra na tabela 'tagsLidas' a data, horário, e o valor de identificação da tag lida. Essa informação pode ser acessada pelo sistema *Web* de gerenciamento. Este comportamento se repete sempre que é realizado alguma modificação na tabela 'livros', ou seja, quando são iniciados os processos de aluguel, devolução, cadastro e/ou remoção de material bibliográfico.

Quando o visitante passar com o material que possui a *tag* de identificação, o leitor irá lê-la e fará uma requisição no BD a fim de obter informações sobre a situação do material. Para realizar essa requisição, o Arduino deve utilizar um módulo *WiFi* para conexão e envio de informações relacionadas a *tag*. Caso a situação do material esteja definida como "alugado", o processo encerra-se. Caso contrário, ou seja, se a situação estiver definida como "não-alugado", o Arduino deve acionar o sistema de auto-falante, aqui representado por um *buzzer*, indicando possível esquecimento de registro de saída de material e/ou tentativa de furto. A Figura 4.4 apresenta o diagrama de sequência no intuito de elucidar o funcionamento geral do *software*.



**Figura 4.4:** Diagrama de Sequência

Para modelagem das tabelas do BD do sistema proposto, a tabela 'livro' foi construída para armazenamento das informações sobre os livros. Cada registro nesta tabela é referente a um livro e contém informações mínimas como o título, a *tag* de identificação e o *status* de alugado ou não alugado.

Foram modeladas também tabelas auxiliares ('solicitacao' e 'tagsLidas') para troca de informações entre a página *Web* e o Arduino. Essas tabelas são utilizadas para permitir com que a página *Web* e o Arduino troquem informações: a coluna 'ler' é a coluna que informa ao Arduino se o servidor requisitou uma leitura de *tag*. O Arduino utiliza a coluna 'tag' para salvamento desta informação no BD. Note na Figura 4.5 a arquitetura das tabelas do BD definidas para apoio no desenvolvimento deste projeto.

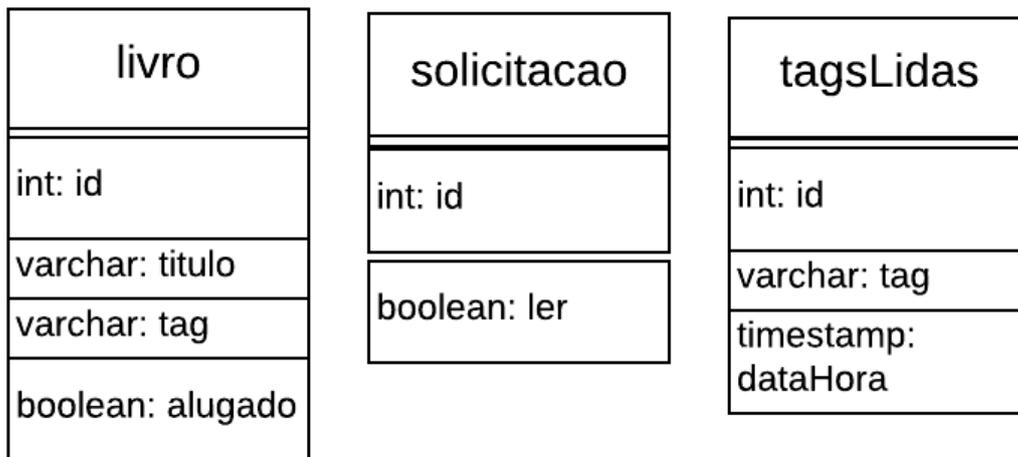


Figura 4.5: Arquitetura do Banco de Dados

### 4.3 Sistema de leitura de *tags* RFID com Arduino

O protótipo do Arduino e seus componentes foram apenas esquematizados e implementados. Contudo, não realizamos a montagem e a validação deste cenário. A Figura 4.6 mostra uma possível prototipação do sistema Arduino que permite a leitura de *tags* RFID, o envio de informações com o uso de um módulo WiFi e o acionamento de um alarme (representado por um *buzzer*).

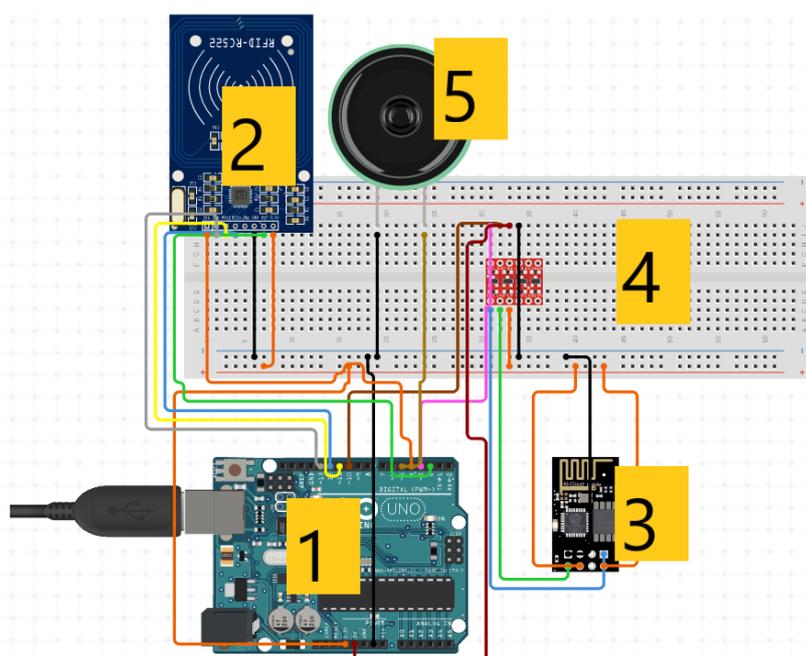


Figura 4.6: Protótipo do Sistema Arduino

### 4.3.1 Descrição dos Componentes de *Hardware*

1. Placa Arduino - O Arduino é a matriz de processamento no qual permite que outros módulos sejam acomodados. É neste componente que se executa os códigos referentes as funcionalidades ofertadas.
2. Sensor RFID - Esse leitor é o componente que permite o processo de leitura de *tags* RFID. O sensor Mfrc522 pode ser utilizado para ser atuar como leitor das *tags*.
3. Módulo *WiFi* ESP8266-01 - O módulo de WIFI ESP8266-01 pode ser utilizado como componente *WiFi* que permite o acesso e conexão à Internet.
4. *Protoboard* - Matriz de contatos que permite conexão entre Arduino e demais componentes.
5. Auto-Falante (*Buzzer*) - Dispositivo capaz de disparar um alarme sonoro.

### 4.3.2 Descrição do Funcionamento do Código para Execução no Arduino

Inicialmente o Arduino executa, uma única vez, o código definido na função padrão *setup()*. Sequencialmente, a função *loop()* é invocada e sua execução é repetida enquanto o Arduino estiver em funcionamento.

Foram definidas 2 (duas) funcionalidades a serem executadas em *loop*: 'VerificaSolicitação()' e 'VerificaSituacao()'. A função 'VerificaSolicitação()' é responsável por pesquisar no banco de dados se existe alguma solicitação de leitura de *tag* ativa. Caso verdadeiro, o Sistema Arduino deve realizar a a leitura da *tag* através do sensor RFID e atualizar o *status* de solicitação de leitura de *tag* no BD.

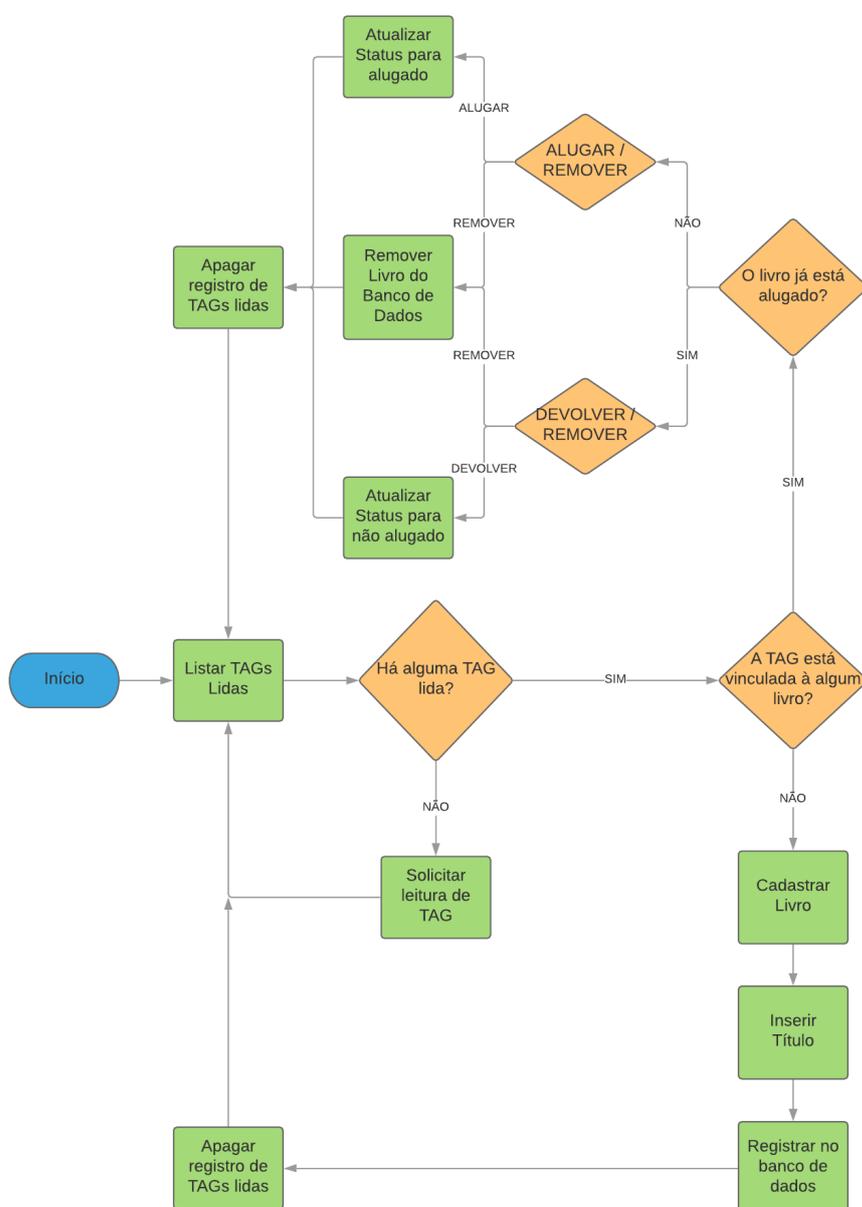
A segunda função que será realizada em *loop*, 'VerificaSituacao()', é a função responsável por verificar a existência de uma *tag* RFID no alcance do sensor de leitura, e caso seja identificada, realiza-se uma consulta desta *tag* lida no BD. Caso a *tag* exista na tabela 'livros' e o *status* for não alugada, dispara-se a função 'TocaAlarme()', indicando o esquecimento de registro de saída de material e/ou furto.

O código completo desenvolvido para o Sistema Arduino pode ser encontrado através da página do projeto<sup>1</sup> hospedado no *GitHub*.

<sup>1</sup><https://github.com/Psherman42000/LivroSec4IFG>

#### 4.4 Página Web

Os processos de interação do usuário com o sistema será somente através da página *Web*. A Figura 4.7 mostra o fluxograma do comportamento e de possibilidades resultantes do desenvolvimento das funcionalidades desenvolvidas, especificadas no documento de requisitos (Apêndice .1). Os códigos referentes ao sistema *Web* também podem ser encontrados na página do projeto<sup>2</sup> hospedado no *GitHub*.



**Figura 4.7:** Fluxograma do Sistema *Web*

<sup>2</sup><https://github.com/Psherman42000/LivroSec4IFG>

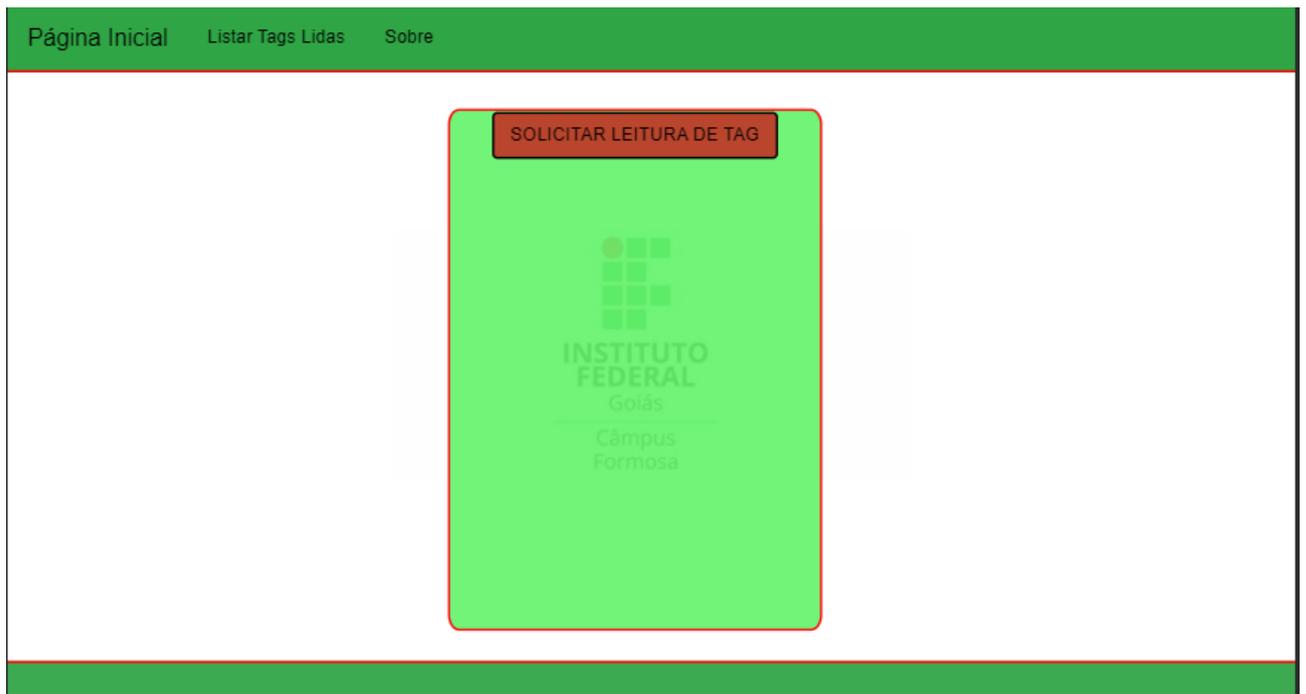
Ao abrir a página *Web* do projeto é mostrada a Página Inicial com uma breve mensagem de introdução ao sistema, conforme ilustrado na Figura 4.8.



**Figura 4.8:** Página inicial do Sistema *Web*

#### 4.4.1 Buscar por Livros

A princípio, o fluxo do sistema inicia-se no momento em que o usuário clica na opção 'Listar *Tags Lidas*'. A Figura 4.9 é o resultado deste processo inicial de execução. Neste local é possível realizar a solicitação ao Arduino para início do procedimento de leitura de *tag*.



**Figura 4.9:** Aba de Listagem de *tags*

Após clicar em solicitar a leitura, o usuário é encaminhado para a página da Figura 4.10 fazendo com que o registro de 'ID' igual a '1' na tabela 'solicitacao' sofra uma alteração na sua coluna 'ler', adotando o valor 1. Quando o Arduino repetir o *loop* de execução, notará a mudança do valor no registro, resultando na realização do registro da próxima *tag* que for identificada pelo leitor RFID na tabela 'tagsLidas'. Feito isso o valor '1' adotado pela coluna 'ler' anteriormente, voltará a assumir o valor '0'. finalizando assim essa solicitação de leitura de *tag*. Se a requisição tiver sido bem sucedida aparecerá essa mensagem ao usuário.



**Figura 4.10:** Mensagem de confirmação para leitura de *tag*

#### 4.4.2 Alugar Livros

Após realizar a solicitação de leitura (listar novamente as *tags* lidas) irá aparecer a *tag* lida pelo Arduino, juntamente com informações pertinentes, tais como horário de leitura e o título do livro a que a *tag* se refere. Logo em seguida, caso o livro não esteja com *status* definido como 'alugado' no BD, o sistema fornecerá as opções de alugar ou remover o material bibliográfico do BD. A Figura 4.11 mostra as informações geradas para possibilitar o início do processo de aluguel.



**Figura 4.11:** Aba de listagem de *tag* após a leitura da *tag*

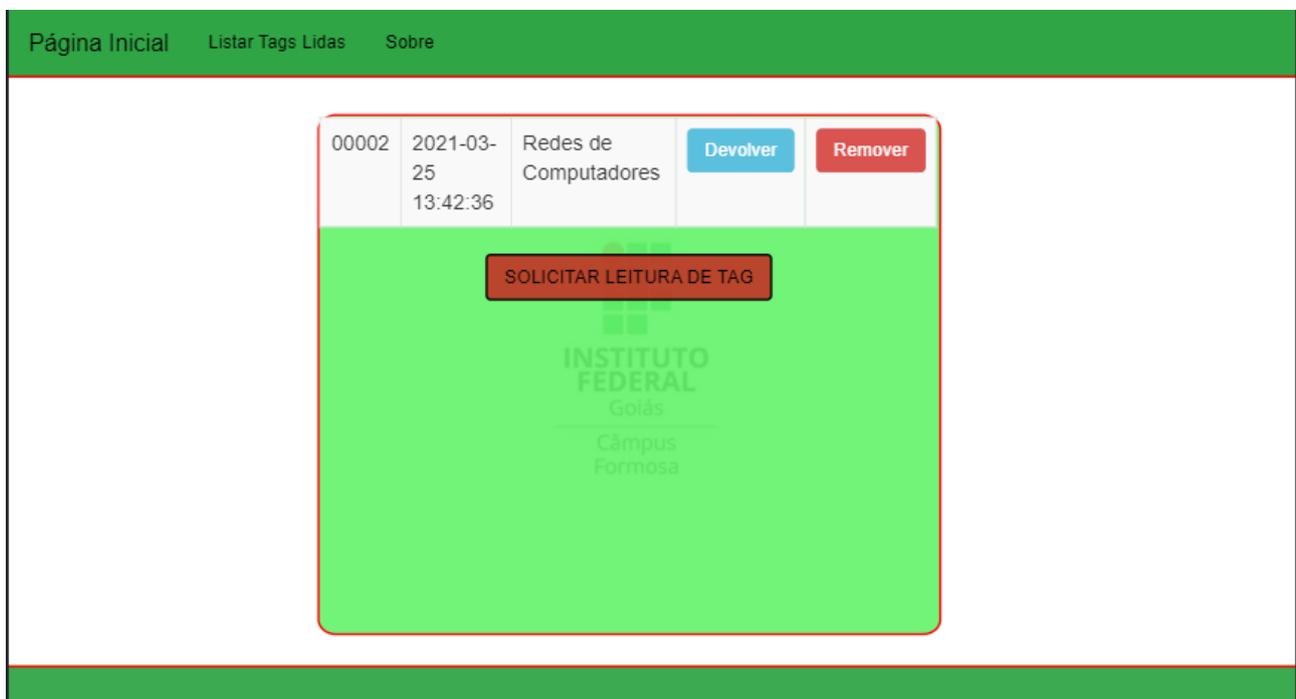
Ao clicar em "Alugar" o *status* de empréstimo do livro é alterado no BD, e o registro da última *tag* lida é removido, conforme pode-se notar na Figura 4.12.



**Figura 4.12:** Mensagem de confirmação de aluguel

### 4.4.3 Devolver Livros

A Figura 4.13 ilustra o cenário de leitura de uma *tag* de um livro que esteja alugado. Neste caso, o sistema oferecerá a opção "Devolver" para possibilitar o devolvimento do material para o acervo bibliográfico.



**Figura 4.13:** Aba de listagem de *tag* quando o livro lido já está alugado

O processo decorrente da funcionalidade 'Devolver' é semelhante ao processo de 'Alugar'. A diferença resume-se no valor de *status* do livro. Neste caso, *status* assumirá valor de não alugado e a mensagem da Figura 4.14 será exibida, confirmando sucesso no procedimento de devolução.



**Figura 4.14:** Mensagem de confirmação de devolução de livro

#### 4.4.4 Cadastrar Livros

Quando o Arduino realizar leitura de uma *tag* inexistente na tabela 'livros', o botão de 'cadastro' é ativado conforme é mostrado na Figura 4.15. Esta página permite a inserção de cadastro desta *tag* como um novo material bibliográfico.



**Figura 4.15:** Aba de listagem de *tag* quando o livro não está cadastrado

Ao iniciar o procedimento de cadastro, o usuário será redirecionado para a página da Figura 4.16, o que possibilita a inserção de informação referente ao título do material associado a *tag* previamente lida.



**Figura 4.16:** Aba de cadastro do livro

Para finalizar o procedimento de cadastro, o sistema *Web* insere informações no BD,

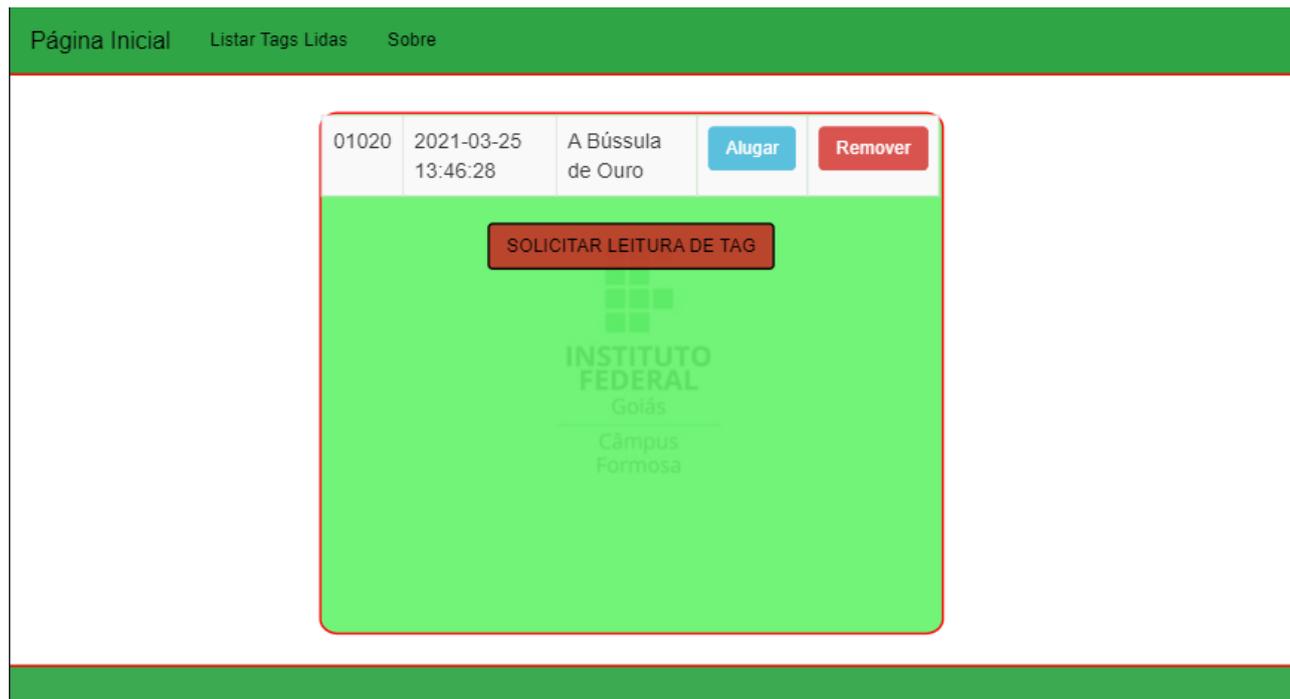
especificamente na tabela 'livro'. A mensagem da Figura 4.17 é exibida confirmando o sucesso no procedimento de cadastro.



**Figura 4.17:** Mensagem de confirmação do cadastro de material bibliográfico

#### 4.4.5 Remover Livros

Caso uma *tag* de um livro seja alterada e/ou danificada, o sistema permite que seja feita a exclusão das informações dos livros e suas respectivas *tags*. O botão 'Remover' inicia o procedimento de exclusão dessas informações, conforme pode ser visto na Figura 4.18. Caso o livro seja devidamente excluído, uma mensagem de confirmação é exibida, semelhantemente ao que acontece em casos de sucesso anteriormente relatados tais como o cadastro de um material.



**Figura 4.18:** Aba de listagem de *tag* após o processo de leitura

## 4.5 Considerações Finais

Neste Capítulo apresentamos os resultados decorrentes do desenvolvimento de um sistema de controle seguro do acervo bibliográfico para o IFG - Câmpus Formosa utilizando a tecnologia RFID e o microcontrolador Arduino, gerenciável por meio de uma aplicação *Web*.

O sistema proposto foi modulado em 2 (dois) subsistemas. O Sistema Arduino foi prototipado para uma possível montagem em *hardware*. Já as funcionalidades em *software* foram modeladas e desenvolvidas.

Um página *Web* foi desenvolvida, a partir da análise (diagramação de alguns artefatos da UML) e implementação dos requisitos identificados. Esta página permite o controle de cadastro de novos materiais bibliográficos, edição da situação atual (alugado ou não alugado) para possibilitar o empréstimo bem como remoção com base na leitura das *tags* de radio frequência identificadas.

# 5

## Conclusão

A popularização dos dispositivos e os avanços da eletrônica permitiram que diversas tecnologias fossem desenvolvidas. Além disso, a queda do preço no *hardware* utilizado em IoT, tais como o Arduino, representam um grande avanço para conexão, comunicação e consolidação de objetos inteligentes e do paradigma de IoT.

Aplicações IoT podem ser desenvolvidas em conjunto com a tecnologia RFID e objetivam trazer maior segurança e controle nos serviços oferecidos. Este trabalho propôs uma solução ao problema de segurança patrimonial do acervo bibliográfico do IFG - Câmpus Formosa ao desenvolver uma solução que, além de controlar o acesso e o empréstimo de material bibliográfico, permite a identificação de possíveis esquecimentos no registro de saída de material e/ou furtos.

O sistema apresentado foi composto por submódulos que em conjunto fornecem solução apropriada para o cadastro de novos materiais bibliográficos, empréstimo e remoção, com base na leitura das *tags* de radio frequência. A implementação proposta neste trabalho possui alto valor agregado pois trás maior segurança no acervo bibliográfico da Instituição, permitindo a devida conservação do patrimônio público.

Como trabalhos futuros pretende-se realizar a implementação do sistema Arduino em *hardware* para possibilitar a integração requerida pelo sistema *Web*, bem como a realização de testes para devida implantação. Outra vertente que pode ser explorada é a integração do sistema minimalista apresentado nesse trabalho com o sistema usado pela biblioteca do IFG - Câmpus Formosa. Essa integração é interessante do ponto de vista da robustez do sistema já em uso e da

não necessidade do uso de diferentes sistemas com propósitos semelhantes, o que minimizaria a curva de aprendizado dos usuários.

## Referências

- ALVES, m. **O que é um Banco de Dados?**, Disponível em: <dicasdeprogramacao.com.br/o-que-e-um-banco-de-dados>. , [S.l.], 2013.
- ANDRADE, a. p. **O que é uma IDE (Ambiente de Desenvolvimento Integrado)?** Disponível em: <www.treinaweb.com.br/blog/o-que-e-uma-ide-ambiente-de-desenvolvimento-integrado/>. , [S.l.], 2020.
- ARAUJO, m. **O Que É Um Microcontrolador?** Disponível em: <edu.ieee.org/br-ufcgras/o-que-e-um-microcontrolador/>. , [S.l.], 2020.
- BALDO, s. **O Modelo relacional**. Unicamp. Disponível em: <www.ic.unicamp.br/geovane/mo410-091/ch03-rm-resumo.pdf>. , [S.l.], p.1–2, 2005.
- BANZI, m. **Primeiros Passos com o Arduino**. Novatec. , [S.l.], v.1, p.3–6, 2011.
- BÔAS, d. **O que é um Sistema Web?** Disponível em: <https://www.villasboasweb.com.br/blog-detalhes.php?id=49>. , [S.l.], 2017.
- CARRION, p. **Internet da Coisas (IoT): definições e aplicabilidade aos usuários finais**. revista human factors design. , [S.l.], v.8, n.15, p.49–66, 2019.
- CARRO, f. r. W. **Sistemas Computacionais Embarcados**. JAI. , [S.l.], p.1, 2003.
- CARVALHO, t. **Internet das Coisas e Sua Aplicação em Bibliotecas**. Revista Gestão.Org. , [S.l.], v.13, p.264–270, 2016.
- CASTRO, a. **APOSTILA: fundamentos de internet das coisas - iot**. faculdades idaam. , [S.l.], v.1, 2019.
- CONVERSE, t. **PHP a Bíblia**. Editora Campus. , [S.l.], v.1, 2003.
- COSTA, a. **RFControl: sistema de gerência de estoque utilizando rfid**. ufop (universidade federal de ouro preto). , [S.l.], 2018.
- CUNHA, a. **o que são sistemas embarcados?** Disponível em: <https://files.comunidades.net/mutcom/artigo\_sist\_emb.pdf>. , [S.l.], 2007.
- FREITAS, r. **CONTROLE E SEGURANÇA PATRIMONIAL POR RFID NO DEPARTAMENTO ACADÊMICO DE ELETRÔNICA DA UTFPR CAMPO MOURÃO**. UTFPR (Universidade Tecnológica Federal do Paraná). , [S.l.], 2020.
- GONÇALVES, m. **APLICAÇÃO DA TECNOLOGIA DE IDENTIFICAÇÃO POR RÁDIOFREQÜÊNCIA (RFID) PARA CONTROLE DE BENS PATRIMONIAIS PELA WEB** Global Scienc and Technology. , [S.l.], v.1, n.7, p.50–59, 2008.
- HIGA, p. **O que é XAMPP e para que serve**. Disponível em: <https://www.techtudo.com.br/dicas-e-tutoriais/noticia/2012/02/o-que-e-xampp-e-para-que-serve.html>. , [S.l.], 2012.
- MOTA, a. **Apostila Arduino Básico**. Vida de Silício. , [S.l.], v.1, 2015.

NASCIMENTO, r. **ALGORITMOS DE LOCALIZAÇÃO DE ETIQUETAS RFID EM BIBLIOTECAS BASEADAS NA INTERNET DAS COISAS**. UFPE (Universidade Federal de Pernambuco). , [S.l.], 2016.

OLIVEIRA, a. **estudo da tecnologia de identificação por radiofrequência - RFID** UnB (Universidade de Brasília). , [S.l.], 2006.

OLIVEIRA, m. **Você precisa saber o que é SQL!** Disponível em: <<https://dicasdeprogramacao.com.br/o-que-e-sql/>>. , [S.l.], 2013.

PEREIRA, f. **Microcontroladores PIC: programação em c**. saraiva educação s.a. , [S.l.], 2007.

PETENATE, m. **identificação por radiofrequência, ou RFID, como funciona esse sistema seguro** Disponível em: <<https://www.escolaedti.com.br/rfid-o-que-e>>. , [S.l.], 2019.

PINHEIRO, j. **Identificação por Radiofrequência: aplicações e vulnerabilidades da tecnologia rfid**. cadernos unifoa. , [S.l.], n.1, 2017.

ROCHA, v. **Sistema de controle de acesso utilizando autenticação por RFID e gerenciamento por meio de software WEB**. UFOP (Universidade Federal de Ouro Preto). , [S.l.], 2019.

SANTOS, b. **Internet das Coisas: da teoria à prática** ufmg (universidade federal de minas gerais). , [S.l.], 2016.

SILVA, l. **INTERNET DAS COISAS: rfid e nfc, conceitos e aplicações** disponível em: <[https://www.academia.edu/36694832/internet\\_das\\_coisas\\_rfid\\_e\\_nfc\\_conceitos\\_e\\_aplica%C3%87%C3%94es](https://www.academia.edu/36694832/internet_das_coisas_rfid_e_nfc_conceitos_e_aplica%C3%87%C3%94es)>. , [S.l.], 2015.

SINGER, t. **Tudo Conectado: conceitos e representações da internet das coisas**. simsocial (2º simpósio em tecnologias digitais sociabilidade). , [S.l.], 2012.

SOARES, j. **REGISTRO DE FREQUÊNCIA DE ALUNOS COM ARDUINO E RFID**. FACIMA (Faculdade da Cidade de Maceió). , [S.l.], 2018.

TAVARES, h. **PLACAS EMBARCADAS E INTERNET DAS COISAS**. FATEC (Faculdade de Tecnologia de Garça). , [S.l.], 2018.

VENTURA, p. **O que é UML (Unified Modeling Language)**. Disponível em: <<https://www.ateomomento.com.br/diagramas-uml/>>. , [S.l.], 2019.

ZURITA, m. **Projeto de Sistemas Embarcados**. UFPI (Universidade Federal do Piauí). , [S.l.], 2011.

# **Apêndice**



**.1 Documento de Requisitos**

---

## **DOCUMENTO DE REQUISITOS**

**Documento de Requisitos versão – 1.4**

**Analista – Pedro Costa Dias.**

**Última Atualização: 07/02/2021**

---

## HISTÓRICO DE REVISÕES DO DOCUMENTO

<b>DATA</b>	<b>VERSÃO</b>	<b>DESCRIÇÃO DA ALTERAÇÃO</b>	<b>AUTOR</b>
27/12/2020	1	CRIAÇÃO DESTE DOCUMENTO.	PEDRO COSTA DIAS
03/01/2021	1.1	ADIÇÃO DE REQUISITOS.	PEDRO COSTA DIAS
10/01/2021	1.2	ADIÇÃO E EDIÇÃO DE REQUISITOS.	PEDRO COSTA DIAS
17/01/2021	1.3	ADIÇÃO DA ESPECIFICAÇÃO DE REQUISITOS.	PEDRO COSTA DIAS
07/02/2021	1.4	REFATORAÇÃO DOS REQUISITOS, EDIÇÃO DAS ESPECIFICAÇÕES DE REQUISITOS.	PEDRO COSTA DIAS

---

## IDENTIFICAÇÃO DOS ENVOLVIDOS

<b>PAPEL</b>	<b>NOME</b>	<b>EMAIL</b>
ANALISTA DE REQUISITOS	PEDRO COSTA DIAS	pedroid199@gmail.com
PRODUCT OWNER	PEDRO COSTA DIAS	pedroid199@gmail.com
DESENVOLVEDOR	PEDRO COSTA DIAS	pedroid199@gmail.com

---

## **PROBLEMA DE NEGÓCIO**

Devido ao fato de não haver uma forma definitiva de prevenção de furtos na biblioteca do IFG – Formosa, é possível que de tempos em tempos haja algum extravio de livros de lá. Visando esse cenário que surgiu a ideia de implementar um sistema antifurto utilizando chips RFID's nos livros e o microcontrolador Arduino para intermediar a gestão desse sistema.

## **REQUISITOS FUNCIONAIS**

**RF001 – A página Web permitir realizar o cadastro de novos livros e suas respectivas tags.**

**RF002 – A página Web deve permitir realizar o aluguel de livros.**

**RF003 – A página Web deve permitir realizar a devolução de livros.**

**RF004 – A página Web deve permitir realizar a exclusão de livros cadastrados no banco de dados.**

**RF005 – A página Web deve poder realizar requisições de leituras de tag.**

**RF006 – O sistema Arduino deve verificar se existem requisições de leitura de tag no banco de dados.**

**RF007 – O Arduino deve realizar a leitura de tag quando aparecerem requisições no banco de dados.**

**RF008 – O Arduino deve salvar a tag lida no banco de dados.**

**RF009 – O Arduino deve emitir um sinal sonoro pelo alto-falante quando um livro não alugado sair pela biblioteca.**

**RF010 – A página Web exibir as tags que foram lidas pelo Arduino.**

---

## REQUISITOS NÃO FUNCIONAIS

RNF001 – O Arduino deve ter acoplado um leitor RFID de longo alcance.

RNF002 – O Arduino deve ter um módulo Wi-Fi conctado para comunicação com o banco de dados.

RNF003 – O Arduino deve ter um alto falante acoplado para emissão de alertas.

RNF004 – Deve haver um banco de dados para armazenar as informações a respeito dos livros.

RNF005 – O site de gerenciamento deve ser acessado somente localmente pelos funcionários da biblioteca.

## ESPECIFICAÇÃO DE REQUISITOS

RF001 - A página Web permitir realizar o cadastro de novos livros e suas respectivas tags.

<b>Referência</b>	RF005 – A página Web deve poder realizar requisições de leituras de tag. RF006 – O sistema Arduino deve verificar se existem requisições de leitura de tag no banco de dados. RF007 – O Arduino deve realizar a leitura de tag quando aparecerem requisições no banco de dados. RF008 – O Arduino deve salvar a tag lida no banco de dados. RF001 - A página Web permitir realizar o cadastro de novos livros e suas respectivas tags.
<b>Sumário</b>	O caso de uso é responsável por realizar o cadastro de uma tag no banco de dados junto com o título do livro que ela representa.

<b>Pré-Condições</b>	O usuário deve realizar a leitura de uma tag que não existe no banco de dados.
<b>Atores</b>	Funcionário da biblioteca.
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O usuário clica “Listar tags lidas”.</li> <li>2. A tag que foi lida pelo Arduino é mostrada junto com o horário da leitura e um botão para “Cadastrar” a tag.</li> <li>3. O usuário clica em “Cadastrar”.</li> <li>4. O usuário é redirecionado para uma página que contém a tag lida, uma caixa para colocar o título, e um botão para “Cadastrar” o livro.</li> <li>5. O usuário clica em “Cadastrar”.</li> <li>6. O sistema registra o novo livro no banco de dados.</li> <li>7. Uma mensagem de confirmação de cadastro realizado aparece para o usuário.</li> </ol>

**RF002 – A página Web deve permitir realizar o aluguel de livros.**

<b>Referência</b>	<p>RF005 – A página Web deve poder realizar requisições de leituras de tag.</p> <p>RF006 – O sistema Arduino deve verificar se existem requisições de leitura de tag no banco de dados.</p> <p>RF007 – O Arduino deve realizar a leitura de tag quando aparecerem requisições no banco de dados.</p> <p>RF008 – O Arduino deve salvar a tag lida no banco de dados.</p> <p>RF001 - A página Web permitir realizar o cadastro de novos livros e suas respectivas tags.</p>
<b>Sumário</b>	O caso de uso é responsável por realizar o aluguel dos livros fazendo a mudança do status dele para “alugado”.

<b>Pré-Condições</b>	O usuário deve realizar a leitura de uma tag esteja com o status como “não-alugado”.
<b>Atores</b>	Funcionário da biblioteca.
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O usuário clica “Listar tags lidas”.</li> <li>2. A tag que foi lida pelo Arduino é mostrada junto com o horário da leitura, botão para “Alugar” o livro e outro para “Remover”.</li> <li>3. O usuário clica em “Alugar”.</li> <li>4. O sistema atualiza o status do livro no banco de dados.</li> <li>5. Uma mensagem de confirmação de aluguel realizado aparece para o usuário.</li> </ol>

**RF003 – A página Web deve permitir realizar a devolução de livros.**

<b>Referência</b>	<p>RF005 – A página Web deve poder realizar requisições de leituras de tag.</p> <p>RF006 – O sistema Arduino deve verificar se existem requisições de leitura de tag no banco de dados.</p> <p>RF007 – O Arduino deve realizar a leitura de tag quando aparecerem requisições no banco de dados.</p> <p>RF008 – O Arduino deve salvar a tag lida no banco de dados.</p> <p>RF001 - A página Web permitir realizar o cadastro de novos livros e suas respectivas tags.</p>
<b>Sumário</b>	O caso de uso é responsável por realizar a devolução dos livros fazendo a mudança do status dele para “não-alugado”.
<b>Pré-Condições</b>	O usuário deve realizar a leitura de uma tag esteja com o status como “alugado”.
<b>Atores</b>	Funcionário da biblioteca.

<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O usuário clica “Listar tags lidas”.</li> <li>2. A tag que foi lida pelo Arduino é mostrada junto com o horário da leitura, botão para “Devolver” o livro e outro para “Remover”.</li> <li>3. O usuário clica em “Devolver”.</li> <li>4. O sistema atualiza o status do livro no banco de dados.</li> <li>5. Uma mensagem de confirmação de devolução realizada aparece para o usuário.</li> </ol>
------------------	--

**RF004 – A página Web deve permitir realizar a exclusão de livros cadastrados no banco de dados.**

<b>Referência</b>	<p>RF005 – A página Web deve poder realizar requisições de leituras de tag.</p> <p>RF006 – O sistema Arduino deve verificar se existem requisições de leitura de tag no banco de dados.</p> <p>RF007 – O Arduino deve realizar a leitura de tag quando aparecerem requisições no banco de dados.</p> <p>RF008 – O Arduino deve salvar a tag lida no banco de dados.</p> <p>RF001 - A página Web permitir realizar o cadastro de novos livros e suas respectivas tags.</p>
<b>Sumário</b>	O caso de uso é responsável por realizar a exclusão permanente das tags e o livro a qual se referem no banco de dados.
<b>Pré-Condições</b>	O usuário deve realizar a leitura de uma tag que já esteja registrada no banco de dados
<b>Atores</b>	Funcionário da biblioteca.

<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O usuário clica “Listar tags lidas”.</li> <li>2. A tag que foi lida pelo Arduino é mostrada junto com o horário da leitura, botão para “Alugar” o livro e outro para “Remover”.</li> <li>3. O usuário clica em “Remover”.</li> <li>4. O sistema Exclui o registro referente a tag da tabela de livros no banco de dados.</li> <li>5. Uma mensagem de confirmação de exclusão realizada aparece para o usuário.</li> </ol>
<b>Alternativas</b>	No passo 2, caso a tag lida esteja com o status “alugado”, um botão para “Devolver” irá aparecer no lugar dele.

**RF005 – A página Web deve poder realizar requisições de leituras de tag.**

<b>Referência</b>	
<b>Sumário</b>	O caso de uso é responsável por registrar no banco de dados que foi feita uma solicitação para o Arduino ler uma tag.
<b>Pré-Condições</b>	O usuário deve estar na página de Listagem de tag.
<b>Atores</b>	Funcionário da biblioteca.
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O usuário clica “Listar tags lidas”.</li> <li>2. Um botão para “Solicitar leitura de tag” deve ser apresentado na tela.</li> <li>3. O usuário clica em “Solicitar leitura de tag”.</li> <li>4. O sistema registra a solicitação no banco de dados.</li> <li>5. Uma mensagem de confirmação de solicitação realizada aparece para o usuário.</li> </ol>

---

**RF006 – O sistema Arduino deve verificar se existem requisições de leitura de tag no banco de dados.**

<ul style="list-style-type: none"><li>• <b>Referência</b></li></ul>	<p>RF006 – O sistema Arduino deve verificar se existem requisições de leitura de tag no banco de dados.</p> <p>RF007 – O Arduino deve realizar a leitura de tag quando aparecerem requisições no banco de dados.</p> <p>RF008 – O Arduino deve salvar a tag lida no banco de dados.</p>
<ul style="list-style-type: none"><li>• <b>Sumário</b></li></ul>	<p>O caso de uso é responsável por sempre estar verificando em loop se a tabela “solicitação” está com um registro de valor que represente “true”.</p>
<ul style="list-style-type: none"><li>• <b>Pré-Condições</b></li></ul>	<p>O Arduino deve estar ligado e conectado à internet.</p>
<ul style="list-style-type: none"><li>• <b>Atores</b></li></ul>	<p>Sistema Arduino.</p>
<ul style="list-style-type: none"><li>• <b>Descrição</b></li></ul>	<ol style="list-style-type: none"><li>1. O sistema conecta-se ao banco de dados.</li><li>2. O sistema consulta as informações da tabela “solicitação”.</li><li>3. O sistema identifica o registro com um valor “true”.</li><li>4. O sistema chama a função de leitura de tag.</li></ol>
<ul style="list-style-type: none"><li>• <b>Alternativas</b></li></ul>	<p>No passo 3, caso o sistema o sistema não identifique o registro com o valor “true”, então não fará nada e voltará pra sua repetição de loop.</p>

**RF007 – O Arduino deve realizar a leitura de tag quando aparecerem requisições no banco de dados.**

<ul style="list-style-type: none"> <li>• <b>Referência</b></li> </ul>	<p>RF006 – O sistema Arduino deve verificar se existem requisições de leitura de tag no banco de dados.</p> <p>RF008 – O Arduino deve salvar a tag lida no banco de dados.</p>
<ul style="list-style-type: none"> <li>• <b>Sumário</b></li> </ul>	<p>Este caso de uso é responsável por ler a próxima tag que leitor RFID identificar.</p>
<ul style="list-style-type: none"> <li>• <b>Pré-Condições</b></li> </ul>	<p>Uma solicitação de leitura deve ter sido identificada no banco de dados.</p>
<ul style="list-style-type: none"> <li>• <b>Atores</b></li> </ul>	<p>Sistema Arduino.</p>
<ul style="list-style-type: none"> <li>• <b>Descrição</b></li> </ul>	<ol style="list-style-type: none"> <li>1. O sistema faz a leitura da próxima tag que o leitor RFID identificar.</li> <li>2. O sistema chama a função para salvar a tag lida no banco de dados.</li> </ol>

**RF008 – O Arduino deve salvar a tag lida no banco de dados.**

<ul style="list-style-type: none"> <li>• <b>Referência</b></li> </ul>	<p>RF006 – O sistema Arduino deve verificar se existem requisições de leitura de tag no banco de dados.</p> <p>RF007 – O Arduino deve realizar a leitura de tag quando aparecerem requisições no banco de dados.</p>
<ul style="list-style-type: none"> <li>• <b>Sumário</b></li> </ul>	<p>Este caso de uso é responsável por salvar a tag que foi lida pelo Arduino na tabela “tagsLidas” no banco de dados.</p>
<ul style="list-style-type: none"> <li>• <b>Pré-Condições</b></li> </ul>	<p>Uma solicitação de leitura deve ter sido identificada no banco de dados, O Arduino deve ter realizado a leitura da tag.</p>
<ul style="list-style-type: none"> <li>• <b>Atores</b></li> </ul>	<p>Sistema Arduino</p>
<ul style="list-style-type: none"> <li>• <b>Descrição</b></li> </ul>	<ol style="list-style-type: none"> <li>1. O sistema fará o registro da tag lida na tabela “tagsLidas” no banco de dados.</li> </ol>

**RF009 – O Arduino deve emitir um sinal sonoro pelo alto-falante quando um livros não alugado sair pela biblioteca.**

<ul style="list-style-type: none"> <li>• <b>Referência</b></li> </ul>	
<ul style="list-style-type: none"> <li>• <b>Sumário</b></li> </ul>	Caso de uso responsável por alertar com um sinal sonoro caso um livro “não-alugado” seja identificado passando pelo leitor.
<ul style="list-style-type: none"> <li>• <b>Pré-Condições</b></li> </ul>	O sistema deve não ter identificado nenhuma solicitação de leitura no banco de dados.
<ul style="list-style-type: none"> <li>• <b>Atores</b></li> </ul>	Sistema Arduino.
<ul style="list-style-type: none"> <li>• <b>Descrição</b></li> </ul>	<ol style="list-style-type: none"> <li>1. O sistema deve ler a tag que passou pelo sensor.</li> <li>2. O sistema consultará o status de empréstimo do livro pela tag lida.</li> <li>3. Se o status do livro estiver como “não-alugado” o sistema ativará o alto-falante.</li> <li>4. O alto-falante irá emitir um sinal sonoro para alertar o caso.</li> </ol>
<ul style="list-style-type: none"> <li>• <b>Alternativas</b></li> </ul>	No passo 3, caso o sistema o status do livro esteja como “alugado”, o sistema não fará nada e voltará pra sua repetição de loop.

**RF010 – A página Web exibir as tags que foram lidas pelo Arduino.**

<ul style="list-style-type: none"> <li>• <b>Referência</b></li> </ul>	<p>RF001 – A página Web permitir realizar o cadastro de novos livros e suas respectivas tags.</p> <p>RF002 – A página Web deve permitir realizar o aluguel de livros.</p> <p>RF003 – A página Web deve permitir realizar a devolução de livros.</p> <p>RF004 – A página Web deve permitir realizar a exclusão de livros cadastrados no banco de dados.</p>
---	--

	<p>RF005 – A página Web deve poder realizar requisições de leituras de tag.</p> <p>RF008 – O Arduino deve salvar a tag lida no banco de dados.</p>
<ul style="list-style-type: none"> <li>• <b>Sumário</b></li> </ul>	<p>Caso de uso responsável por realizar a exibição na página de listagem de tags da tag que foi lida e salva no banco de dados pelo Arduino, juntamente com seus respectivos botões com suas funcionalidades, sendo eles aluguel, devolução, cadastro e remoção.</p>
<ul style="list-style-type: none"> <li>• <b>Pré-Condições</b></li> </ul>	<p>O Arduino deve ter lido e salvado a tag no banco de dados.</p>
<ul style="list-style-type: none"> <li>• <b>Atores</b></li> </ul>	<p>Página Web.</p>
<ul style="list-style-type: none"> <li>• <b>Descrição</b></li> </ul>	<ol style="list-style-type: none"> <li>1. O usuário deve clicar em “Listar tags lidas”.</li> <li>2. O sistema fará uma consulta na tabela “tagsLidas” do banco de dados.</li> <li>3. O sistema exibirá as a tag e o horário em que ela foi lida.</li> <li>4. O sistema exibirá os botões que a tag tem possibilidade de realizar (“Alugar” /“Devolver” /“Cadastrar” /“Remover”).</li> <li>5. O sistema também exibirá o botão de “Solicitar leitura de tag”.</li> </ol>