

Daniel Zonta Ojeda  
Deivid Márlon Fernandes da Costa  
Gabriel de Souza Padilha

**Ctrl P**  
**Controle de Patrimônios**

Foz do Iguaçu

2019

Daniel Zonta Ojeda  
Deivid Márlon Fernandes da Costa  
Gabriel de Souza Padilha

## **Ctrl P**

# **Controle de Patrimônios**

Documentação do projeto para gerenciamento de patrimônios criado pelos alunos do curso de graduação de ciências da computação da UNIOESTE do ano de 2019 na disciplina de Introdução a Ciências da Computação.

Universidade do Oeste do Paraná  
UNIOESTE

Foz do Iguaçu  
2019

# Resumo

Este relatório está sendo criado para documentar o modo de utilização do sistema de controle patrimonial CtrlP. O protótipo utiliza um Raspberry Pi 3 vinculado a um sensor de radio frequência(RFID), capaz de registrar o momento em que uma tag está próxima. Este sistema de sensor e tag será utilizado para monitorar a movimentação de patrimônios uma empresa, e também irá atualizar automaticamente os dados registrados em tempo real. Os dados registrados serão salvos no banco de dados online *Firebase* e estarão disponíveis para visualização para os administradores do sistema.

**Palavras-chave:** RFID. gerenciamento de patrimônios. raspberry pi 3.

# Abstract

This report is being created to document the way to use the patrimony control system by CtrlP. The prototype uses a Raspberry Pi 3 linked to a radiofrequency sensor (RFID), able to register the moment where the tag is close. This tag-sensor system will be used to monitor the movimentation of patromny in a company. The prototype will automatically record the data in real time. The data will be stored in the online database *Firebase* and will be available to the system administrator.

**Keywords:** RFID. patromony management. raspberry pi 3.

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>5</b>
<b>2</b>	<b>SOBRE O PROJETO</b>	<b>6</b>
<b>3</b>	<b>INSTRUÇÕES DE USO</b>	<b>7</b>
<b>3.1</b>	<b>Tipos de login</b>	<b>7</b>
<b>3.2</b>	<b>Administrador</b>	<b>7</b>
3.2.1	Criando usuários	7
3.2.2	Gerenciando patrimônios	8
3.2.3	Gerenciando ocorrências	8
<b>3.3</b>	<b>Acesso de manutenção</b>	<b>8</b>
<b>4</b>	<b>ESPECIFICAÇÕES TÉCNICAS</b>	<b>9</b>
<b>4.1</b>	<b>Hardware</b>	<b>9</b>
4.1.1	Raspberry	9
4.1.2	Sensor RFID	9
4.1.3	Tag RFID	10
4.1.4	Capa	10
<b>4.2</b>	<b>Software</b>	<b>11</b>
4.2.1	Firebase	11
4.2.2	Scripts	11
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>13</b>
<b>5.1</b>	<b>Problemas na biblioteca da firebase</b>	<b>13</b>
<b>5.2</b>	<b>Melhorias futuras</b>	<b>13</b>
5.2.1	Sensor RFID	13
5.2.2	Leitura de códigos de barras	13
	<b>REFERÊNCIAS</b>	<b>14</b>

# 1 Introdução

Reduzir os custos, e, por consequência, aumentar o capital, é um dos maiores desafios enfrentados por uma empresa. Em meio a competitividade do mercado, o ideal seria não comprometer a qualidade do produto ou serviço prestado. Portanto, deve-se buscar reduzir os custos de operação.<sup>[1]</sup>

Os bens de uma empresa, em conjunto, representam um valor significativo. Razão pela qual é preciso adotar métodos que protejam este capital. Tal método precisa:

- Ser confiável;
- Ser transparente;
- Informar aos usuários onde determinados patrimônios estão;
- Permitir à administradores adicionar, remover e editar os dados de patrimônios;
- Permitir que usuários possam registrar ocorrências aos patrimônios.
- Não necessitar de entrada do usuário para continuamente rastrear o objeto;
- Ser financeiramente viável;

Uma possível solução para lidar com estes problemas, é o RFID (*Radio Frequency Identification*).<sup>[2]</sup> Utilizando sensores RFID, conectados a uma Raspberry Pi, é possível criar um sistema que atenda a todos estes requisitos e que seja de baixo custo. Tal custo torna o sistema acessível para empresas de todo porte, melhorando seu fluxo de equipamento e tornando o ambiente de trabalho funcional e otimizado.

## 2 Sobre o projeto

O controle patrimonial pode ser entendido como o gerenciamento de todo o controle patrimonial de uma empresa, servindo para prevenir complicações para o fluxo de caixa.<sup>[3]</sup> Devido aos patrimônios constituírem capital passivo, algumas adversidades como a ausência de registros na movimentação e extravios acabam gerando divergências administrativas e na contabilidade. Ter um sistema de controle patrimonial possibilita a empresa reduzir gastos e otimizar processos de compras.

Porém, realizar o controle manual de um elevado número de patrimônios, registrando cada movimentação e alteração é uma tarefa exaustiva, divergindo recursos que poderiam serem aplicados em outras áreas. Nos resta então, automatizar este processo.

A automatização de tal controle patrimonial irá potencialmente gerar menos custos e melhor aproveitamento do tempo, por meio da integração de sistemas e da substituição de atividades humanas por automáticas.<sup>[4]</sup> Uma tecnologia viável para realizar esta automatização são sensores RFID ligados à uma Raspberry Pi.

Para o rastreamento dos objetos ocorrer, o objeto a ser rastreado deve receber uma etiqueta (ou tag) eletrônica de RFID, e para que a sua leitura e identificação aconteçam, devem ser instaladas leitores de RFID no local onde o item será acondicionado.<sup>[5]</sup> Muitos tipos de tags RFID existem, podendo serem divididas em duas classes: ativas e passivas. As tags são compostas de uma antena, um semicondutor acoplado à antena e uma capsula. No caso das tags ativas, é necessário também a fonte de energia.<sup>[6]</sup> Uma grande inovação recente foi o emprego de tags passivas de RFID tornando-se, assim, uma solução de baixo custo, podendo ser amplamente utilizado por diversos segmentos do mercado.<sup>[5]</sup> Estas serão as tags utilizadas, já que é inviável utilizar uma bateria para cada objeto a ser rastreado.

## 3 Instruções de uso

### 3.1 Tipos de login

O sistema pode possuir três tipos diferentes de login: administrador, funcionário e usuário público. As diferenças são:

O administrador:

- Pode editar, adicionar e remover usuários;
- Pode editar, adicionar e remover patrimônios;
- Pode editar, adicionar e remover ocorrências;
- Tem acesso a movimentação de patrimônios;

O funcionário é a pessoa cuja responsabilidade é a de verificar e resolver ocorrências.

- Pode visualizar ocorrências;
- Pode marcar uma ocorrência como resolvida;

E o usuário público é classificado como qualquer pessoa com acesso à empresa, tendo a opção de criar ocorrências caso exista algum problema com um patrimônio

- Pode criar ocorrências;

### 3.2 Administrador

#### 3.2.1 Criando usuários

O menu de usuários nos dá opções para criar e listar usuários.

Para criar um usuário, basta selecionar a opção "Criar Usuário" no menu de usuários, e digitar as informações necessárias.

A lista de usuários fornece Nome, Email e Departamento de todos os usuários cadastrados, como indicado na figura ??.



### 3.2.2 Gerenciando patrimônios

O menu dos patrimônios nos dá opções para criar e editar patrimônios.

Para criar um novo patrimônio, basta selecionar "Criar Patrimônio", e então fornecer o número e a descrição do patrimônio.

Para editar um patrimônio, selecione "Edição de Patrimônios" no menu de patrimônios, e então selecionar da lista de patrimônios qual será editado, como ilustrado na figura ??.

### 3.2.3 Gerenciando ocorrências

O menu ocorrências contém 3 opções: Criar ocorrência, Minhas ocorrências e Gestão de ocorrências.

Para criar uma nova ocorrência, é necessário o número do patrimônio e uma descrição da ocorrência.

Minhas ocorrências gera uma lista de todas as ocorrências, seus patrimônios e sua situação, conforme ilustrado na figura ??.

A gestão de ocorrências nos dá uma lista mais detalhada de todas as ocorrências, separadas pela sua situação, conforme figura ??.

## 3.3 Acesso de manutenção

O login de manutenção dá acesso ao menu "Minhas tarefas", onde o funcionário pode checar as ocorrências pendentes e atendê-las.

## 4 Especificações Técnicas

### 4.1 Hardware

#### 4.1.1 Raspberry

O modelo da Raspberry utilizada é a Raspberry Pi 3 Model B,, ilustrada na figura ???. Ela possui as seguintes especificações técnicas:<sup>[7]</sup>

- Quad Core 1.2 GHz Broadcom BCM2837 64bit CPU;
- 1GB Ram;
- BCM43438 wireless LAN e Bluetooth Low Energy (BLE) on board;
- 100 Base Ethernet;
- 40-pin extended GPIO;
- 4 USB 2 ports;
- 4 Polos com saída estéreo e compositor de vídeo;
- Entrada HDMI;
- Entrada Micro SD;

#### 4.1.2 Sensor RFID

Para realizar a leitura das tags RFID, será utilizado um módulo leitor RFID baseado no chip MFR522, da empresa NXP. Este chip, de baixo consumo e pequeno tamanho, permite sem contato ler e escrever em cartões que seguem o padrão Mifare, muito usado no mercado.<sup>[8]</sup>

Especificações:<sup>[8]</sup>

- Corrente de trabalho: 13-26mA / DC 3.3V;
- Corrente ociosa: 10-13mA / 3.3V;
- Corrente Slep: <80uA – Pico de corrente: <30mA;
- Frequência de operação: 13,56MHz;
- Tipos de cartões suportados: Mifare1 S50, S70 Mifare1, Mifare UltraLight, Mifare Pro, Mifare Desfire;

- Temperatura de operação: -20 a 80 graus Celsius;
- Temperatura ambiente: -40 a 85 graus Celsius;
- Umidade relativa: 5% – 95%;
- Parâmetro de Interface SPI;
- Taxa de transferência: 10 Mbit/s;
- Dimensões: 8,5 x 5,5 x 1,0cm;
- Peso: 21g;

#### 4.1.3 Tag RFID

As tags utilizadas serão tags passiva do tipo cartão. O cartão possui um número de identificação (UID) pré-gravado e memória de 1KB para armazenar dados, embora estes não sejam utilizados.

Especificações:<sup>[9]</sup>

- Frequência de operação: 13,56MHz;
- Material: PVC (Prova d'água);
- Alcance: 10mm;
- Padrão: Mifare S50;
- Memória: 1K byte EEPROM (768 bytes livres);
- Durabilidade de escrita: 100.000 ciclos;
- Padrão ISO: ISO 14443 / 14443A;
- Dimensões: 85 x 54 x 0,9mm;
- Peso: 7g;

#### 4.1.4 Capa

Para a segurança do sistema, servindo tanto para proteger a Raspberry contra condições atmosféricas, quanto para protegê-la de uso indevido, como por exemplo, ter um pen drive com arquivos maliciosos inserido na entrada USB, foi desenvolvida uma capa protetora, obstruindo todas as entradas da Raspberry, com a exceção da fonte de energia. A seguir, seguem os desenhos técnicos desta capa:

## 4.2 Software

### 4.2.1 Firebase

Para salvar as operações na nuvem, a tecnologia utilizada é o banco de dados Firebase do Google. Esta plataforma já nos provê com funcionalidades como análises, relatórios e autenticação de usuários. A estrutura do banco de dados é a seguinte:

### 4.2.2 Scripts

Cada Raspberry representa um node, e deve existir um node na entrada de todo recinto onde se deseja gravar movimentações de patrimônios.

No momento em que a Raspberry é ligada, um script *python* é executado utilizando *systemd*, enviando o IP do node para a firebase, possibilitando os administradores a acessarem a Raspberry via *ssh*. Por padrão, o usuário é "pi" e a senha é "raspberrypi". Por exemplo, se o endereço ip for 192.168.1.2, o administrador pode ter acesso à Raspberry com:

```
$ ssh pi@192.168.1.2
```

O script executado na inicialização é:

```
import requests
import os
import time

nodeName="-Lv28cBFqV0r570agXUH"
while (True):
    time.sleep(60)
    os.system("ip addr > /home/pi/ipaddr.txt")
    def getIpOnFile():
        with open('/home/pi/ipaddr.txt', 'r') as file:
            for line in file:
                if "inet" in line and "brd" in line:
                    ip = line.split(" ")[5]
                    return ip
    def check_internet():
        url='http://www.google.com/'
        timeout=100
        try:
            _ = requests.get(url, timeout=timeout)
            return True
        except requests.ConnectionError:
            print("Sem conexão")
            return False
    config = {
        "apiKey": "AIzaSyBWp_iVUt4e_2PDHJic5z2DKWbjWbvKOOI",
        "authDomain": "patri-control.firebaseio.com",
        "databaseURL": "https://patri-control.firebaseio.com",
        "projectId": "patri-control",
        "storageBucket": "patri-control.appspot.com",
        "messagingSenderId": "629490990593",
        "appId": "1:629490990593:web:bba3ec7c6e7538de"
    }
```

```

while(check_internet() == True):
    firebase = pyrebase.initialize_app(config)
    banco = firebase.database()
    banco.child().update({"nodes/"+nodeName+"/ip": getIpOnFile()})
    getIpOnFile()
os.system("rm /home/pi/ipaddr.txt")

```

Ou seja, após 60 segundos são decorridos (para garantir que a Raspberry tem conexão à Internet) um arquivo temporário "ipaddr.txt" é criado, contendo o conteúdo do comando bash *ip addr*. Uma conexão com o Firebase é criada, e, caso haja internet, apenas a área correspondente ao ip do arquivo ipaddr.txt é enviada à Firebase, sob seu respectivo node.

Para que a leitura das tags comece, é necessário que um administrador execute o arquivo start.py, localizado na pasta home:

```
$ python3 ~/start.py
```

Grande parte deste script foi adaptado de PiMyLifeUp<sup>[10]</sup>. Ao executar o programa, o seguinte irá acontecer:

```

targetPatrimonio_name, text = reader.read()
targetPatrimonio_key = ''

```

A leitura do patrimônio é realizada.

```

snapshotPatrimonio = banco.child("patrimonios").order_by_child("name").equal_to(
(targetPatrimonio_name).get())
if (snapshotPatrimonio.val()):
    targetPatrimonio_key = snapshotPatrimonio.each()[0].key()

```

Verifica se esse patrimônio já está presente no node.

```

if (snapshotPatrimonio.each()[0].val()['node']==nodeName):
    snapshot = banco.child("nodes/"+nodeKey+"/patrimonios").order_by_child("name").
equal_to(targetPatrimonio_name).get()# parar de deletar tudo
    banco.child("nodes/"+nodeKey+"/patrimonios/"+snapshot.each()[0].key()).remove()
    banco.child().update({"patrimonios/"+targetPatrimonio_key+"/node":"sem node"})
else:
    if (snapshotPatrimonio.each()[0].val()['node']!='sem node'):
        snapshotNodeAntigo=banco.child("nodes").order_by_child("name").equal_to
(snapshotPatrimonio.each()[0].val()['node']).get()
        nodeAntigoKey = snapshotNodeAntigo.each()[0].key()
        snapshot = banco.child("nodes/"+nodeAntigoKey+"/patrimonios").
order_by_child("name").equal_to(targetPatrimonio_name).get()# parar de deletar tudo
        banco.child("nodes/"+nodeAntigoKey+"/patrimonios/"+snapshot.each()[0].
key()).remove()

```

Caso o patrimônio esteja no node, ele é retirado, e caso o patrimônio não esteja no node, ele é adicionado.

## 5 Considerações Finais

### 5.1 Problemas na biblioteca da firebase

Na execução do projeto, foi encontrado um problema com o arquivo "pyrebase.py" da biblioteca da firebase. Esse problema causava erro ao gravar dados do tipo string no banco de dados. Para resolver este problema, foi necessário editar o arquivo:

```
/usr/lib/python3.5/site-packages/pyrebase/pyrebase.py
```

É necessário substituir a linha 238:

```
parameters[param] = quote('' + self.build_query[param] + '')
```

Por

```
parameters[param] = '' + self.build_query[param] + ''
```

Este processo funciona até o momento (dezembro de 2019).

### 5.2 Melhorias futuras

#### 5.2.1 Sensor RFID

Os leitores RFID do tipo portal poderiam serem utilizados. Sensores deste tipo permitem centenas de operações por segundo, nos assegurando com segurança que todos as tags serão lidas. Se os sensores forem posicionados em todas as entradas, teremos segurança que a movimentação de todo patrimônio que passe pela entrada será registrada. Sua desvantagem está em seu preço, cerca de 1000% maiores que os sensores utilizados atualmente. Um exemplo de sensor do tipo portal está ilustrado na figura ??

#### 5.2.2 Leitura de códigos de barras

Poderia ser interessante para o sistema a possibilidade da leitura de códigos de barra em conjunto com o sistema RFID, já que muitas empresas já utilizam o código de barras no controle de seus patrimônios. Esta inclusão poderia garantir uma transição mais harmoniosa do controle patrimonial antigo para o controle patrimonial da CtrlP.

# Referências

- 1 COPASTUR BLOG. [S.l.], 2016. Disponível em: <<https://www.copastur.com.br/blog/10-maneras-de-reduzir-custos-operacionais-na-empresa/>>. Acesso em: 26 Nov 2016. Citado na página 5.
- 2 NARCISO, M. G. Aplicação da tecnologia de indentificação por rádiofrequência (rfid) para controle de bens patrimoniais pela web. *Global Science and Technology*, v. 1, n. 7, p. 50–59, Dec. - Mar. 2008. Citado na página 5.
- 3 FRAZÃO, M. *O que é controle patrimonial? Conceito, como fazer e porque fazer*. [S.l.], 2017. Disponível em: <<https://investorcp.com/gestao-ativo-imobilizado/o-que-e-controle-patrimonial/>>. Acesso em: 29 Nov 2019. Citado na página 6.
- 4 RIBEIRO, M. *Automatizar processos, veja como a Pluga pode te ajudar!* [S.l.], 2016. Disponível em: <<https://pluga.co/blog/gestao-empresarial/automatizar-processos/>>. Acesso em: 29 Nov 2019. Citado na página 6.
- 5 RFID BRASIL. *Sistema de localização em tempo real: quais as aplicações?* [S.l.], 2017. Disponível em: <<https://rfidbrasil.com/blog/sistema-de-localizacao-em-tempo-real-quais-as-aplicacoes/>>. Acesso em: 29 Nov 2019. Citado na página 6.
- 6 WANT, R. An introduction to rfid technology. *IEEE pervasive computing*, IEEE, n. 1, p. 25–33, 2006. Citado na página 6.
- 7 SITE OFICIAL RASPBERRY. *Raspberry Pi Model B*. [S.l.], 2019. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>. Acesso em: 30 Nov 2019. Citado na página 9.
- 8 FILIPE FLOP. *Kit Módulo Leitor RFID MFC522 Mifare*. [S.l.], 2019. Disponível em: <<https://www.filipeflop.com/produto/kit-modulo-leitor-rfid-mfrc522-mifare/>>. Acesso em: 30 Nov 2019. Citado na página 9.
- 9 FILIPE FLOP. *Cartão Rfid Programável Mifare 13,56Mhz*. [S.l.], 2019. Disponível em: <<https://www.filipeflop.com/produto/cartao-rfid-programavel-mifare-1356mhz/>>. Acesso em: 30 Nov 2019. Citado na página 10.
- 10 PIMYLIFEUP. *How to setup a Raspberry Pi RFID RC522 Chip*. [S.l.], 2017. Disponível em: <<https://pimylifeup.com/raspberry-pi-rfid-rc522/>>. Acesso em: 15 Mai 2019. Citado na página 12.