Universidade Tecnológica Federal do Paraná – UTFPR
Departamento Acadêmico de Eletrônica – DAELN
Departamento Acadêmico de Informática – DAINF
Engenharia de Computação
Oficina de Integração 3 (EEX23) – S71 – 2018/1

---

# Relatório Técnico
# **Safe Place**

Augusto Domingos augustodomingos@alunos.utfpr.edu.br
Bruna de Paula Monarin bruna@alunos.utfpr.edu.br
Débora Sandi deborasandi@alunos.utfpr.edu.br
José Reinaldo Lopes da Silva joses@alunos.utfpr.edu.br
Leandro dos Santos Ferreira lferreira@alunos.utfpr.edu.br

Junho de 2018

**Resumo**

A segurança de ambientes laboratoriais e empresariais é um fator crucial não apenas para a seguridade de dados, informações e bens materiais, como também para a vida de todas as pessoas que ocupam esses espaços. Sabendo disso, o presente relatório apresenta o desenvolvimento do Safe Place, um sistema de controle de acesso de pessoas a ambientes restritos automatizado. Este tem o intuito de mitigar falhas na segurança na entrada, fator muito comum nos sistemas de controle de acesso convencionais, como no caso de uso de roletas e cartões de identificação. Para isso, o espaço a ser protegido pelo sistema é separado fisicamente do ambiente externo por um cubículo, onde são realizadas autenticações para verificar a legitimidade dos usuários. O sistema possui quatro etapas de autenticação distintas: senha numérica, mensuração de peso, biometria da impressão digital e reconhecimento facial. Além disso, caso ocorra alguma situação de ameaça, um alarme é acionado e contatos de emergência são contatados via aplicativo de mensagens instantâneas. Todo o gerenciamento do sistema, é realizado pelo seu administrador via aplicativo Web. Por fim, para a realização da prova de conceito, foi elaborado um protótipo em miniatura para a comprovação do seu funcionamento.

## 1 Introdução

Sabendo que a inexpugnabilidade de ambientes é um fator inalcançável, empresas de segurança e/ou controle de acesso visam cada vez mais aprimorar as técnicas e tecnologias empregadas em seus sistemas. Por meio de dispositivos capazes de inviabilizar a concretização de violações, procura-se reduzir ao máximo danos causados por terceiros, uma vez que apenas o uso de analistas e pessoas treinadas para a vigília de ações delituosas pode ser insuficiente, por muitas vezes corruptível [1].

O presente documento aborda o desenvolvimento de um sistema de controle de acesso de pessoas a ambientes altamente restritos, visando assegurar o acesso apenas de pessoas autorizadas. Este é compreendido por duas barreiras físicas, cada uma contendo uma porta de correr, empregadas para delimitar três zonas: não protegida, de autenticação e segura. Para garantir esse controle existem quatro tipos de autenticação: senha numérica, mensuração de peso, leitura de impressão digital e reconhecimento facial do indivíduo.

Por se tratar de um sistema de controle de acesso, é imprescindível a utilização de um sistema de alarme. Portanto, em situações de risco ou emergência, ou mesmo tentativa de coerção para entrada no ambiente, o sistema de alarme é ativado, acarretando no disparo de uma sirene e no envio de mensagens, via Telegram (aplicativo de mensagens instantâneas), para os contatos de emergência previamente cadastrados pelo administrador do sistema. O gerenciamento do sistema, como cadastramento de usuários e registros de acessos é realizado através de um aplicativo *Web* hospedado em uma nuvem. Já o interfaceamento de todos os componentes é realizado através do Raspberry Pi, como apresentado na Figura 1.

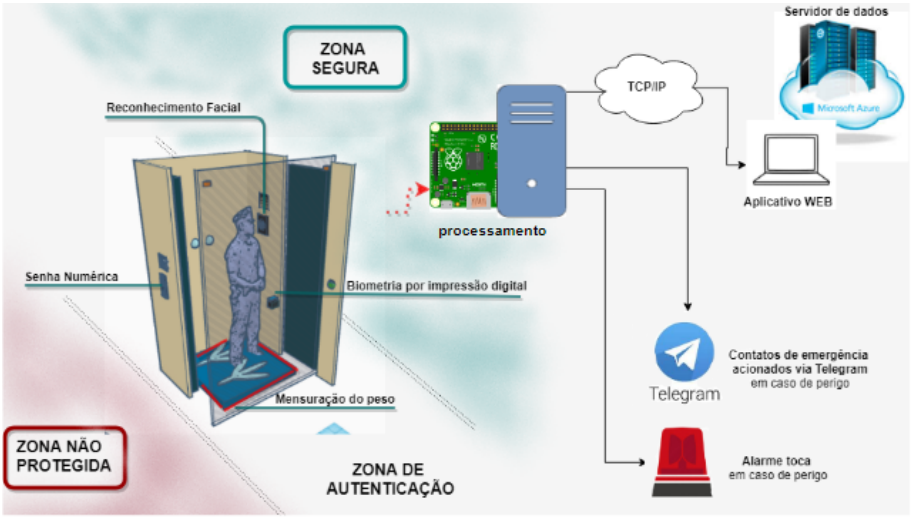



Figura 1: Visão geral do sistema - Safe Place.

## 1.1 Requisitos funcionais

Para a execução do projeto foram estabelecidos requisitos funcionais, utilizados para especificar o comportamento do sistema. A Tabela 1 apresenta os requisitos funcionais do sistema, separados por seu respectivos tópicos: cadastro, controle de acesso e segurança.

Tabela 1: Requisitos Funcionais do sistema.

| | |
|---|---|
| | **CADASTRO** |
| **RF01** | O sistema deverá permitir o cadastramento de pessoas. |
| **RF01.1** | O cadastro deverá permitir ao usuário cadastrar as seguintes informações: nome, função, senha de acesso e senha de emergência, impressão digital, foto da face e peso. |
| **RF01.2** | O sistema deverá permitir a exclusão de cadastros. |
| **RF01.3** | O sistema deverá permitir a atualização de cadastros. |
| | **CONTROLE DE ACESSO** |
| **RF02** | O sistema deverá permitir entrada e saída de usuários através do acionamento das portas. |
| **RF02.1** | Para a entrada, o sistema deverá permitir ao usuário inserir senha e ser reconhecido por sua impressão digital, reconhecimento facial e peso. |
| **RF02.2** | Para a saída, o sistema deverá permitir ao usuário apertar um botão interno ao ambiente restrito para abrir a segunda porta e passar a impressão digital para abrir a primeira porta. |
| | **SEGURANÇA** |
| **RF03** | O sistema deverá permitir a entrada de uma senha de segurança que avise o administrador que há problemas. |
| **RF03.1** | A senha de segurança deve acionar a abertura da primeira porta. |
| **RF03.2** | Após 30 segundos uma sirene será acionada no interior do ambiente restrito. |
| **RF03.3** | Uma foto deve ser tirada e enviada ao administrador assim que a porta for aberta. |
| **RF04** | O sistema deverá tentar reconhecer a face do usuário em cinco tentativas. |
| **RF04.1** | Caso o usuário não seja reconhecido, o sistema deverá abrir a porta 1. |
| **RF05** | O sistema deverá tentar reconhecer a impressão digital do usuário em três tentativas. |
| **RF05.1** | Caso o usuário não seja reconhecido, o sistema deverá abrir a porta 1. |
| **RF06** | O sistema deverá acionar a sirene quando alguma das portas tiver seu fechamento impedido. |
| **RF07** | O sistema deverá acionar a sirene quando a balança detectar a presença de um peso não reconhecido por 60 segundos. |

## 1.2 Requisitos Não Funcionais

Além dos requisitos funcionais citados anteriormente, foram elaborados também os requisitos não funcionais do sistema. Estes dizem respeito a como as funcionalidades serão entregues ao usuário final e são apresentados na Tabela 2, separados por seus respectivos tópicos: protótipo e comunicação.

Tabela 2: Requisitos Não Funcionais do sistema.

| | |
|---|---|
| | **PROTÓTIPO** |
| **RNF01** | A maquete do projeto mostrará o funcionamento das portas em proporção 1:10 em comparação com o tamanho real. |
| **RNF02** | A balança não será projetada para suportar 2 pessoas. |
| **RNF03** | O sistema não garantirá a segurança do usuário. |
| **RNF04** | O espaço entre as duas portas deve ser limitado a largura da balança. |
| **RNF05** | O sistema deverá integrar um gancho para que o usuário coloque seus pertences. |
| **RNF06** | O sistema deverá ter duas portas em linha reta para permitir o acesso ao ambiente restrito. |
| **RNF06.1** | A primeira barreira deve ser ultrapassada por meio de inserção de senha individual. |
| **RNF06.2** | A segunda barreira deve ser ultrapassada a partir da checagem de impressão digital, peso do usuário e reconhecimento facial. |
| **RNF07** | As portas do protótipo serão de correr. |
| **RNF08** | O material utilizado para o protótipo será papel Paraná. |
| | **COMUNICAÇÃO** |
| **RNF09** | O sistema embarcado e a estação base devem se comunicar por cabo Ethernet. |

# 2 Dispositivos utilizados

## 2.1 Sistema Embarcado

Dentro do escopo do projeto, buscou-se encontrar um sistema embarcado capaz de suprir e permitir o gerenciamento de toda a gama de periféricos acoplados ao composto. Almejando assim, suportar não apenas os periféricos responsáveis pelas várias etapas do processo de autenticação, como também os encarregados por tornar a experiência do usuário mais agradável, intuitiva e completa. Como é o caso do uso de caixas de som e display, que retornam *feedbacks* ao utilizador durante o cumprimento de todas as etapas do processo. Optou-se então pela utilização do *Raspberry Pi 3 Model B*, por cobrir as necessidades locais do projeto [2]. A seguir estão descritas as principais especificações referentes a esse modelo:

1. 1.2GHz 64-bit *quad-core* ARMv8 CPU
2. 1GB RAM
3. 4 portas USB
4. 40 pinos de entrada/saída
5. Porta *Ethernet*
6. Saída de áudio (pino de 3.5mm)
7. Interface para câmera Raspicam(CSI)

Por todas as especificações previamente apresentadas, este microcontrolador supre as necessidades do projeto, sendo elas: processador veloz para integrar todas as funcionalidades do sistema, porta *Ethernet* necessária para a comunicação com o servidor de processamento, 40 pinos de entrada e saída para a conexão de células de carga, sensores magnéticos, botão, sirene, ponte H, relés e *display OLED*. Além disso, há o interfaceamento padrão com a câmera *RaspiCam* para o reconhecimento de pessoas, e entradas *USB* para teclado numérico, alimentação de caixa de som e leitor de impressão digital. A Figura 2 apresenta o esquema das conexões do *Raspberry Pi* com os dispositivos supracitados (desprezando-se as fontes e pinos de alimentação).

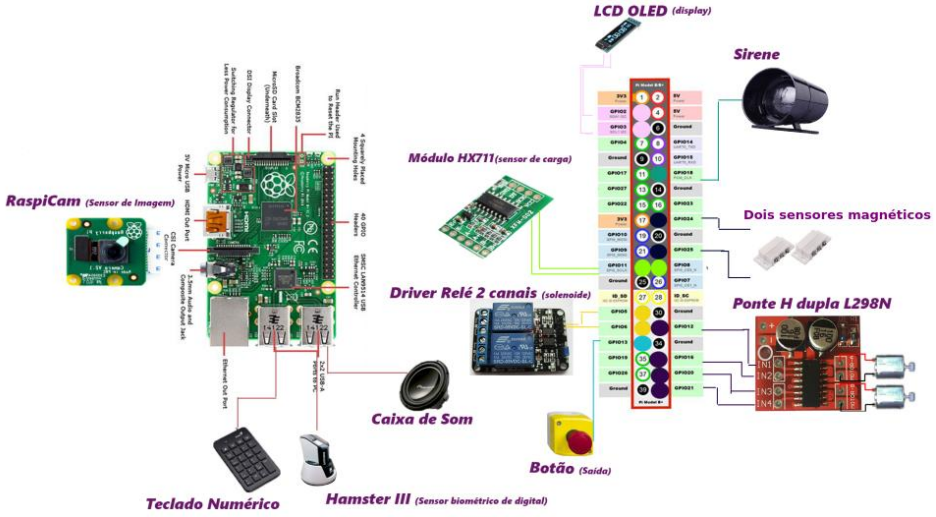



Figura 2: Esquemático das ligações Raspberry Pi.

## 2.2 Comunicação

O sistema embarcado se comunica com o servidor de processamento, também considerado a estação-base do projetosendo que este por sua vez se comunica com o servidor de dados. O primeiro - que se encontra alocado em um computador e conectado via cabo ethernet com o sistema embarcado - é o responsável pela coleta de dados oriundos do sistema embarcado, pelo processamento dessas informações e pela comunicação com o segundo servidor. Já o segundo, é composto por um banco de dados *MySQL* hospedado e acessível via interface *WEB* pelo *phpMyAdmin* e é utilizado para armazenar todos os dados de cadastros e acessos de usuários. Os protocolos de comunicação entre o sistema embarcado e a estação-base, e entre a estação-base e o servidor de dados, estão apresentados na Figura 3.

A comunicação é realizada utilizando o protocolo de comunicação *TCP*, uma vez que é necessário assegurar que os pacotes sejam recebidos corretamente na


Figura 3: Esquemático das ligações Raspberry Pi.

ordem certa.

# 3 Funcionamento

## 3.1 Cadastro

O primeiro passo para a utilização do sistema, é o cadastramento dos usuários que terão acesso ao ambiente protegido. Esse cadastro é realizado por meio de um aplicativo *Web*, ao qual apenas o administrador possui acesso, e contém os seguintes dados dos usuários: nome, CPF, gênero, data de nascimento, cargo, email, telefone, senha, impressão digital, peso e foto. Para a obtenção dos dados de autenticação, o administrador deve inicialmente inserir a senha de cadastro de novo usuário no teclado numérico, para iniciar o procedimento. Então, o novo usuário deve digitar a senha desejada no teclado numérico, assim como subir na balança para mensurar seu peso atual, colocar três dedos distintos no leitor biométrico e por fim ter sua foto de rosto tirada pela *RaspiCam*. Os demais dados são então preenchidos no aplicativo *Web* pelo administrador, para finalização do cadastro.

## 3.2 Entrada na Zona Segura

Quando um usuário do sistema deseja entrar na Zona Segura, ele deve passar por todas as quatro etapas de autenticação descritas a seguir.

### 3.2.1 Etapa de autenticação 1

A primeira etapa a ser cumprida pelo usuário é a inserção da sua senha previamente cadastrada, por meio de um teclado numérico localizado ao lado da Porta 1, na Zona Não Protegida (Figura 4-a). Há duas possíveis senhas a serem inseridas: a senha padrão de acesso ou a senha de segurança. Embora ambas permitam que a Porta 1 conceda acesso à zona de autenticação, apenas a senha de acesso permite que as próximas verificações sejam efetuadas, enquanto a senha de segurança é utiliza em situações de emergência ou perigo. Esta faz com que o sistema envie uma mensagem de texto utilizando o aplicativo Telegram, para contatos de emergência previamente cadastrados pelo administrador, para que estes estejam cientes da situação e tomem as devidas providências. Embora

a Porta 1 abra nessa situação, as próximas verificações não poderão ser efetudas, pois o usuário não foi identificado no sistema. O intuito de permitir o acesso à zona de autenticação é delongar a situação até que as medidas necessárias sejam tomadas, tendo em vista que o sistema foi desenvolvido apenas garantir a integridade da zona segura e das pessoas em que nela se encontram.

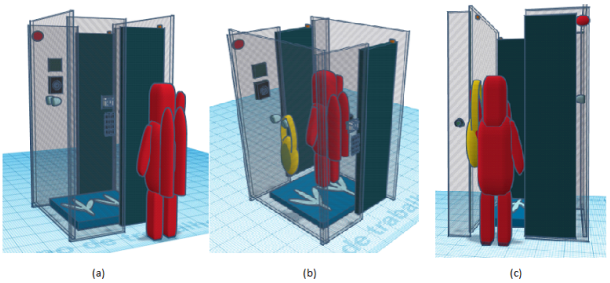



Figura 4: Etapas de Autenticação

### 3.2.2 Etapa de autenticação 2

A segunda etapa é a mensuração do peso, efetuada por meio de células de carga (balança), situadas na zona de autenticação (Figura 4-b). Nessa etapa é necessário que o usuário esteja com a menor quantidade possível de adereços, para que não haja uma divergência superior à 10% de seu peso cadastrado. Caso o usuário esteja carregando algum item pesado, este item deve ser depositado no gancho. Esta necessidade é informada pelo alto-falante após a primeira tentativa de validação do peso ser mal sucedida. Caso o peso mensurado não seja validado após um período de 60 segundos, o sistema infere que há uma tentativa de invasão por um indivíduo desconhecido e efetua o procedimento de emergência. O alarme é então acionado e o envio de mensagem é realizado, como citado na primeira etapa. Caso validado, a Porta 1 fecha, liberando a próxima etapa de autenticação.

### 3.2.3 Etapas de autenticação 3 e 4

A terceira e quarta etapa consistem na leitura biométrica de impressão digital e reconhecimento facial do usuário, permitindo 3 e 5 tentativas, respectivamente. Assim como na etapa 2, a falha em qualquer uma dessas autenticações ocasionam o acionamento do procedimento de emergência, com exceção da senha numérica. Caso ambas sejam validadas, a Porta 2 é aberta, permitindo o acesso do usuário à Zona Segura (Figura 4-c). O diagrama de atividades da Figura 5 apresenta as etapas para entrar na Zona de Autenticação, de forma processual.

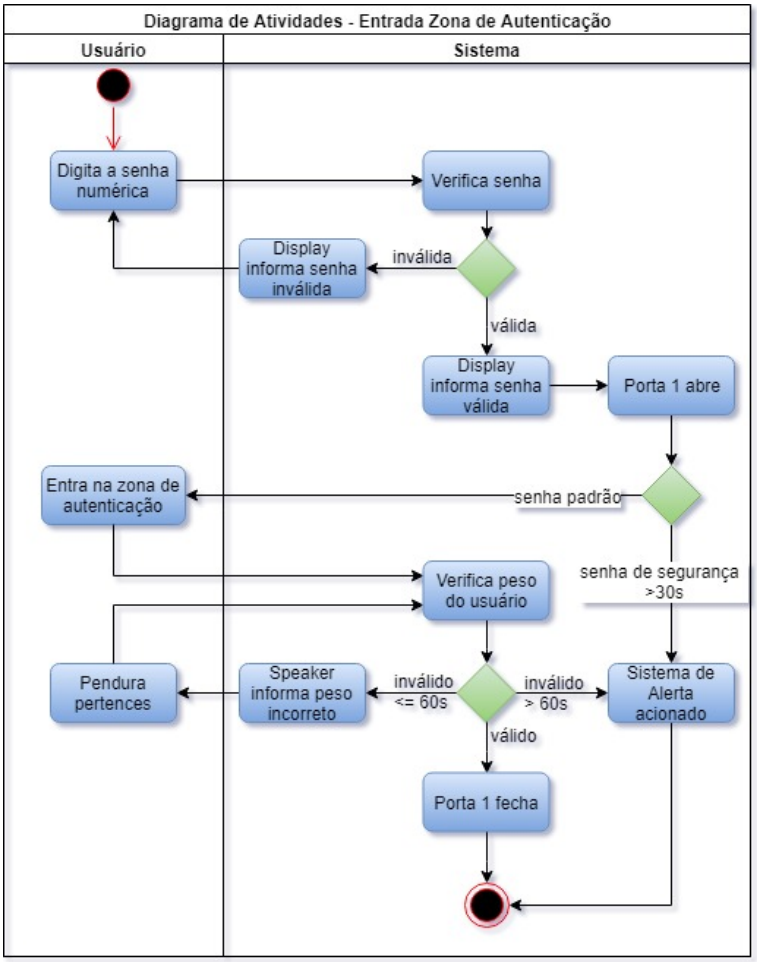



Figura 5: Diagrama de Atividades - Entrada.

# 4 Elaboração

## 4.1 Protótipo

Para a construção da prova de conceito do projeto foi desenvolvido um protótipo da Zona de Autenticação de proporção 1:10. Para tanto, foram utilizadas duas folhas de Papel Paraná n. 60, recortadas com estilete e colocadas com cola branca para junção e fixação. Após a montagem, o protótipo foi pintado com tinta guache branca para acabamento. A Figura 6, apresenta duas visões distintas do protótipo construído.

(a) (b)

Figura 6: Protótipo.

## 4.2 Células de Carga

A segunda etapa de autenticação, visa garantir que apenas um indivíduo esteja presente dentro da zona de autenticação através da mensuração do peso. A ideia é que as células de carga delimitem todo o espaço, de forma que nenhum espaço do chão fique sem monitoramento. Busca-se, através da avaliação do peso, evitar episódios de *tailgating* e *piggybacking*. *Tailgating* é o termo utilizado para descrever a situação em que uma pessoa não autorizada procura entrar num ambiente ao qual não tem autorização "indo atrás"de alguém autorizado, sem o consentimento desta. Já *piggybacking* é utilizado para descrever essa situação quando ocorre com o consentimento da pessoa autorizada. Ou seja, a mensuração do peso busca impedir que pessoas não autorizadas, de forma consentida ou não, sigam um usuário. Para a prova de conceito, optou-se pela utilização de uma balança digital, que possui quatro células de carga acopladas.

Seu funcionamento é baseado no extensômetro (*strain gauge*) alojado no interior de cada uma das células. Ao aplicar uma força externa sobre ela, ele sofre uma deformação de seu fio resistivo e por consequência o valor da resistência é alterado. A medição dos pesos em cima da célula, nesta aplicação, é feita pela tensão, já que ela muda de acordo com a resistência do fio.

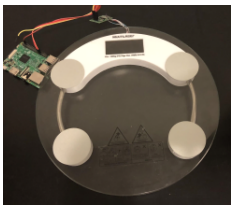


Figura 7: Balança utilizada no projeto.

A leitura e conversão do sinal da tensão na balança é feita pelo módulo HX711. O esquemático da ligação das células de carga ao módulo está representado na Figura 8.

A montagem do circuito entre as células da balança e o módulo HX711 foi

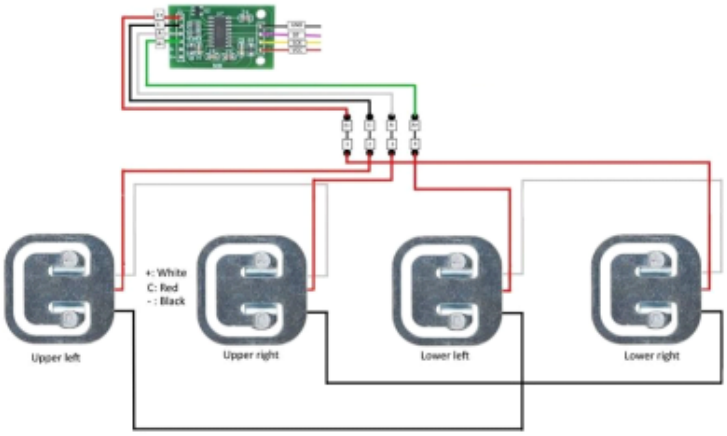



Figura 8: Esquemático de ligação das células de carga ao módulo HX711.

feita através da própria placa da balança digital. Nela pode-se soldar os quatros fios de leitura do módulo (como mostrado na Figura 8) e fazer a conexão com as células, para obtenção dos dados da conversão do sinal analógico para digital.[3]

## 4.3 Leitor de impressão digital

O leitor biométrico escolhido para o projeto (Fingkey Hamster III) é um produto comercial cuja biblioteca não é open source. Seu funcionamento é realizado através de *drivers* e bibliotecas instaladas no sistema. A programação foi feita a partir do manual da *API* que fornece as funções necessárias para a inicialização do dispositivo, captura de digitais e comparação. A maior parte dos dados manipulados foi realizado por objetos de classes específicos da *enBSP SDK*, pacote que contém as funções disponibilizadas para o programador. Optou-se pela não utilização da memória interna do dispositivo por questões de segurança. Ao invés disso, é utilizado um banco de dados para armazenar os dados cadastrais. A Figura 9 apresenta o leitor biométrico utilizado no projeto.

Figura 9: Leitor biométrico Fingkey Hamster III utilizado no projeto.

## 4.4 Reconhecimento Facial

O reconhecimento facial foi baseado no TCC do aluno Edgar Teixeira, orientando do Professor Heitor Silvério Lopes [4]. O autor original criou um sistema

de reconhecimento facial a partir de redes neurais o qual foi adaptado para o presente projeto.

As Redes Neurais Artificiais (RNAs) são algoritmos bioinspirados fundamentados na natureza para criar uma solução para determinado problema. Com base no funcionamento cerebral, as redes neurais seguem a ideia do conjunto de neurônios para classificação. Compostas por diversas unidades interconectadas, denominadas neurônios, estas possuem a capacidade de aprender a realizar uma certa tarefa, dado um conjunto de exemplos. Cada uma dessas unidades apresenta um comportamento específico de entrada/saída determinado por três fatores: pela função de transferência; pelas interconexões com outras unidades; e pelas entradas externas [5].

O diferencial do método utilizado é o seu funcionamento: independente da posição da pessoa, se estiver com barba, boné, óculos de grau entre outros, é possível reconhecer a "identidade facial" do usuário. O reconhecimento ocorre em quatro etapas:

1. Detecção de faces: o algoritmo (rede neural treinada) tenta encontrar a localização das faces na imagem.
2. Alinhamento: ajuste da posição do rosto na imagem – centralização e rotação com o objetivo de enquadrar o rosto.
3. Reconhecimento facial: cálculo de um vetor a partir da imagem fornecida utilizando rede neural treinada com milhares de fotos de famosos – encontram-se os "landmarks" da face e calculam-se distâncias.
4. Comparação: compara-se o vetor criado com o cadastrado para o usuário utilizando um limite de 0.6 para cada posição. Caso haja compatibilidade de pelo menos 99%, considera-se a mesma pessoa.

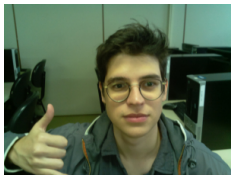
Foram realizados diversos testes com pessoas de diferentes etnias e cores de cabelos. Verificou-se que a luz do ambiente e a qualidade da câmera influenciam no resultado; por exemplo, uma pessoa de pele clara com cabelos claros e uma luz muito forte pode não ser reconhecida. Os testes com pessoas de cabelo moreno e pele clara retornam melhores resultados.

As Figuras 10, 11 e 12 apresentam exemplos de testes realizados e seus respectivos resultados.

(a) (b)

Figura 10: Teste 1: usuário com e sem óculos de grau.

Resultado: 100% de acurácia.


(a)


(b)

Figura 11: Teste 2: usuário não olha para a foto; usuário com a cabeça inclinada.

Resultado: 99,22% de acurácia.


(a)

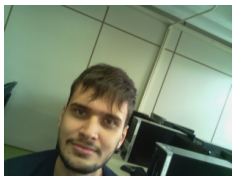

(b)

Figura 12: Teste 3: usuário com diferente corte de cabelo; foto com inclinação.

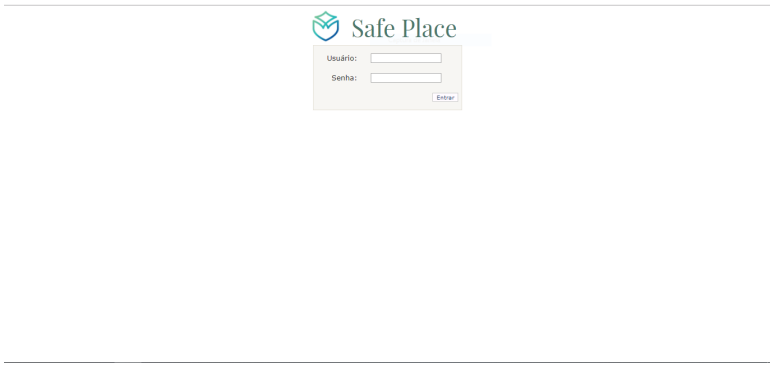
Resultado: 100% de acurácia.

### 4.5 Aplicativo WEB

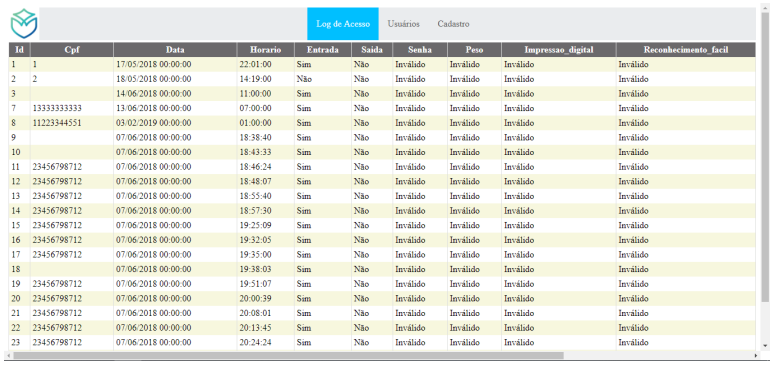
Para o gerenciamente de usuários e logs do sistema, foi desenvolvido um aplicativo *Web* utilizando o *framework* ASP.NET [6], que permite o desenvolvimento de aplicativos web, serviços e *websites* dinâmicos. Por se tratar de código compilado, o uso dessa ferramenta pode tornar a execução de uma aplicação mais rápida do que se esta fosse desenvolvida em uma linguagem interpretada, como por exemplo PHP. As aplicações ASP.NET utilizam linguagens orientadas à objetos, como C++, C# ou VB.NET, podendo até reutilizar códigos de diferentes linguagens pertencentes à projetos desenvolvidos para a plataforma .NET [7]. Além disso, para facilitar o acesso à esse aplicativo, o mesmo foi publicado em uma máquina virtual no servido em nuvem Microsoft Azure.

O aplicativo desenvolvido é composto por 4 telas, sendo elas:

1. Tela de *login*, na qual é necessário que o administrador insira seu *login* e senha, para que haja o acesso às demais funcionalidades do aplicativo (Figura 13a).
2. Tela de *log*, que permite visualizar os dados referentes à entrada, saída e tentativas falhas de acesso ao ambiente restrito (Figura 13b).
3. Tela de listagem de usuários, pra visualização dos dados de todos os usuários cadastrados no sistema (Figura 13c).
4. Tela de cadastro, que contém todas as informações de usuários à serem inseridas no sistema (Figura 13d).

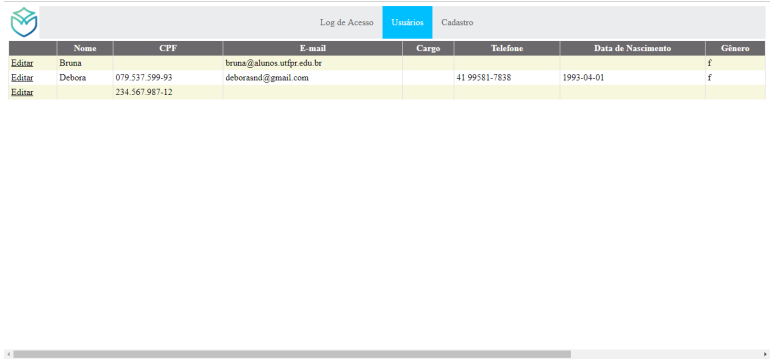



(a) Tela de *Login*.



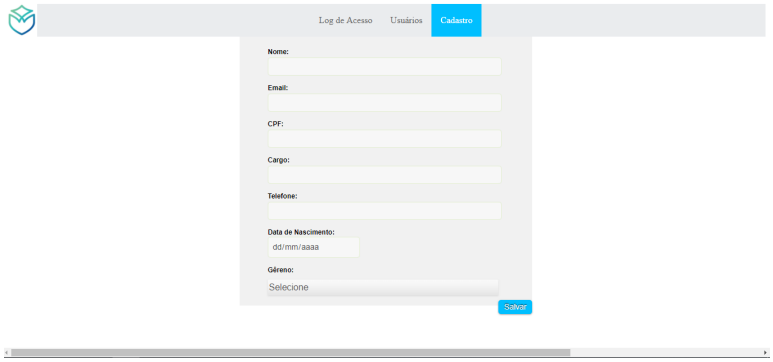

| Id | Cpf | Data | Horario | Entrada | Saida | Senha | Peso | Impressao_digital | Reconhecimento_facil |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 17/05/2018 00:00:00 | 22:01:00 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 2 | 2 | 18/05/2018 00:00:00 | 14:19:00 | Não | Não | Inválido | Inválido | Inválido | Inválido |
| 3 | | 14/06/2018 00:00:00 | 11:00:00 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 7 | 13333333333 | 13/06/2018 00:00:00 | 07:00:00 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 8 | 11223344551 | 03/02/2019 00:00:00 | 01:00:00 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 9 | | 07/06/2018 00:00:00 | 18:38:40 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 10 | | 07/06/2018 00:00:00 | 18:43:33 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 11 | 23456798712 | 07/06/2018 00:00:00 | 18:46:24 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 12 | 23456798712 | 07/06/2018 00:00:00 | 18:48:07 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 13 | 23456798712 | 07/06/2018 00:00:00 | 18:55:40 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 14 | 23456798712 | 07/06/2018 00:00:00 | 18:57:30 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 15 | 23456798712 | 07/06/2018 00:00:00 | 19:25:09 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 16 | 23456798712 | 07/06/2018 00:00:00 | 19:32:05 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 17 | 23456798712 | 07/06/2018 00:00:00 | 19:35:00 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 18 | | 07/06/2018 00:00:00 | 19:38:03 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 19 | 23456798712 | 07/06/2018 00:00:00 | 19:51:07 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 20 | 23456798712 | 07/06/2018 00:00:00 | 20:00:39 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 21 | 23456798712 | 07/06/2018 00:00:00 | 20:08:01 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 22 | 23456798712 | 07/06/2018 00:00:00 | 20:13:45 | Sim | Não | Inválido | Inválido | Inválido | Inválido |
| 23 | 23456798712 | 07/06/2018 00:00:00 | 20:24:24 | Sim | Não | Inválido | Inválido | Inválido | Inválido |

(b) Tela de *Log*.




| | Nome | CPF | E-mail | Cargo | Telefone | Data de Nascimento | Gênero |
|---|---|---|---|---|---|---|---|
| Editar | Bruna | | bruna@alunos.utfpr.edu.br | | | | f |
| Editar | Debora | 079.537.599-93 | deborasnd@gmail.com | | 41 99581-7838 | 1993-04-01 | f |
| Editar | | 234.567.987-12 | | | | | |

(c) Tela de Usuários.




(d) Tela de Cadastro.

### 4.6 Base de Dados

A linguagem SQL (Structured Query Language) é utilizada para manipular bancos de dados relacionais através de SGBDs (Sistema de Gerenciamento de Banco de Dados), permitindo a execução de diversas operações, como inserir e alterar registros, consultar informações, entre outros. MySQL é um SGBD de código aberto que utiliza a linguagem SQL.

A criação da estrutura da base de dados iniciou-se com a criação de um diagrama com as tabelas necessárias e seus atributos, apresentado na Figura 14. Em seguida, a criação das tabelas foi programada em linguagem SQL e executada no SGBD MySQL, hospedado no site phpMyAdmin.

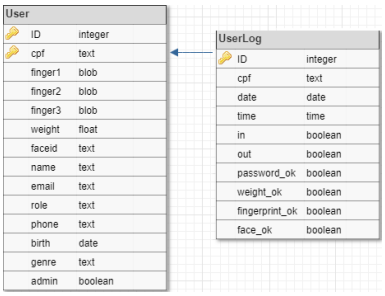



Figura 14: Tabelas do banco de dados.

### 4.7 Telegram

O aviso de invasão, ameaça ou situação de risco no ambiente é enviado via Telegram, aplicativo de mensagens instantâneas, para um grupo de contatos de emergência cadastrado pelo administrador. É utilizado para o envio da mensagem a *API*, que gerencia o *bot* do Telegram, desenvolvida na linguagem de programação Python.

Foi criado então um *bot* próprio para o sistema, através da *Bot API* do Telegram, chamado *TheSafePlaceBot*. O *bot* é um sistema capaz de interagir com o usuário em uma determinada aplicação para simular ações humanas. No caso do projeto, foi criado e configurado pelo assistente criador de bots da própria plataforma *BotFather*, através do aplicativo móvel do Telegram.[8]

As interações do *TheSafePlaceBot* com o usuário no sistema são especificadas pelo *software* em Python, o qual realiza a comunicação com o servidor do Telegram e envia os dados necessários. No presente caso, é utilizado para envio de mensagem de texto e uma foto tirada pela RaspiCam.

## 5 Orçamento

A Tabela 3 apresenta a lista de materiais utilizados no desenvolvimento projeto, com seus respectivos valores.

Tabela 3: Orçamento

| Produto | Preço |
|---|---|
| Raspberry Pi 3 | R$ 160,00 |
| Câmera RaspiCam | R$ 75,00 |
| Balança | R$ 49,90 |
| 2 Motores DC | R$ 30,00 |
| Ponte H | R$ 25,00 |
| Leitor Biométrico | R$ 400,00 |
| Teclado numérico c/ Fio | R$ 20,00 |
| Sirene | R$ 12,00 |
| Botão | R$ 1,00 |
| Sensores magnéticos | R$ 40,00 |
| Materiais protótipo | R$ 50,00 |
| Display OLED | R$ 20,00 |
| 2 solenoides eletromagnéticas | R$ 30,00 |
| 2 Fontes AC/DC | R$ 30,00 |
| Mini Alto-falante | R$ 20,00 |
| Módulo conversor HX711 | R$ 20,00 |
| Cabo extensor 50cm câmera | R$ 23,43 |
| Total | R$ 1006,33 |

# 6 Considerações finais

No decurso da realização do projeto abordado, foi assumido o desafio de unificar algumas das diversas áreas lecionadas no decorrer do curso de Engenharia de Computação da UTFPR câmpus Curitiba. Por conseguinte, nos foi permitido empregar muitos dos conhecimentos adquiridos em áreas como: sistemas inteligentes, banco de dados, redes de computadores, eletrônica geral, sistemas embarcados, engenharia de software, gestão de pessoas, análise e projeto de sistemas e desenvolvimento Web.

Dentre os diversos riscos previstos durante a especificação do projeto, dois dos dez considerados os mais prováveis de ocorrer realmente foram confrontados: a ausência temporária de um membro da equipe; e problemas com os componentes eletrônicos. Apesar da ausência provocar a necessidade da redistribuição de tarefas, ainda assim foi possível manter o avanço de acordo com o cronograma especificado, em virtude da colaboração de todos os outros membros. Devido à problemas encontrados com alguns dos componentes eletrônicos utilizados, foi necessário a utilização da reserva financeira já prevista para a substituição dos mesmos.

Pode-se então, com esse projeto, verificar a real importância de um planejamento prévio especificado detalhadamente, contendo as principais diretrizes

do projeto. Bem como a estipulação de soluções pré estabelecidas para possíveis problemas a serem encontrados.

## Agradecimentos

Ao aluno Edgar Teixeira, do curso de Engenharia de Computação da UTFPR câmpus Curitiba, por ceder o algoritmo de reconhecimento facial desenvolvido em seu TCC, para a utilização no projeto. E principalmente aos Professores Doutores João Alberto Fabro e Heitor Silvério Lopes pela orientação e auxílio no decorrer do semestre.

## Referências

[1] Marco Antonio dos Santos; Jacinto Rodrigues Franco. Inteligência para gestores de segurança e mediação de conflitos. https://www.passeidireto.com/arquivo/20140976/inteligencia-par-agestores-de-seguranca-e-mediacao-de-conflitos.

[2] RASPBERRY PI DOCUMENTATION. `https://www.raspberrypi.org/documentation/hardware/`.

[3] Build a digital Raspberry Pi Scale (with Weight Sensor HX711). `https://tutorials-raspberrypi.com/digital-raspberry-pi-scale-weight-sensor-hx711/`.

[4] Edgar J. Teixeira. Desenvolvimento de um sistema de reconhecimento facial para controle de acesso. Universidade Tecnológica Federal do Paraná., 2017.

[5] Leandro N. de Castro; Fernando J. Von Zuben. Redes Neurais Artificiais. vm1 dca.fee.unicamp.br/pub/docs/vonzuben/ia006_03/topico5_03.pdf.

[6] Learn ASP.NET. `https://www.asp.net/learn`.

[7] Create a Web API with ASP.NET Core and Visual Studio for Windows. `https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-2.1`.

[8] Telegram Bots: An introduction for developers. `https://core.telegram.org/bots`.