

**UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS MISSÕES -
CAMPUS DE ERECHIM
DEPARTAMENTO DE ENGENHARIAS E CIÊNCIA DA COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

MURILO FERANTI

**GEOPOINT - APLICATIVO PARA GERENCIAMENTO DE PONTO DIGITAL PARA
AGÊNCIAS**

**ERECHIM - RS
2021**

MURILO FERANTI

**GEOPOINT - APLICATIVO PARA GERENCIAMENTO DE PONTO DIGITAL PARA
AGÊNCIAS**

**Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do grau de Bacharel,
Departamento de Engenharias e Ciência
da Computação da Universidade Regional
Integrada do Alto Uruguai e das Missões -
Campus de Erechim.**

Orientador: Prof. MSc. Daniel Menin Tortelli

ERECHIM - RS

2021

AGRADECIMENTOS

Agradeço inicialmente a minha mãe, Eli Maria Cantele pelo empenho depositado em toda minha jornada sempre mostrando a importância dos estudos para minha formação. Pelo investimento na educação, sendo ela dentro ou fora de uma instituição de ensino.

Grato também pelo auxílio na minha proposta de trabalho feito pelo professor do curso, MSc. Daniel Menin Tortelli, orientador do meu projeto. Obrigado pelas orientações e compreensão durante o processo de desenvolvimento.

Agradeço à minha namorada, Daniela Hübner da Silva, que esteve ao meu lado me incentivando para a finalização dessa importante etapa em meus estudos, mostrando sempre que momentos difíceis iriam ser superados. Além disso, me auxiliou com boas práticas de escrita para um melhor desenvolvimento do presente trabalho.

Agradeço também à Universidade Regional Integrada do Alto Uruguai e das Missões - URI Erechim e o seu corpo docente que estiveram sempre comprometidos com a excelência do ensino.

Não importa o que aconteça, continue a nadar.
(WALTERS, Graham)

RESUMO

O gerenciamento de registros de horário ponto é fundamental para o controle do tempo de trabalho dos funcionários de uma empresa. Com a utilização de um sistema anoso, os funcionários acabam perdendo um tempo relevante, além de possíveis esquecimentos ou falhas em seus registros. O presente trabalho consiste no desenvolvimento de um aplicativo por meio da utilização do *framework Ionic*, banco de dados *PostgreSQL* e *API* desenvolvida em *PHP*, que é focado em todo o sistema de registros executados pelos funcionários da empresa. Como resultado, o aplicativo suscitou um meio mais prático utilizando a geolocalização, para verificar se o funcionário encontra-se dentro do seu ambiente de trabalho para efetuar o registro automaticamente e, assim, sanando o problema de custo de tempo e de possíveis esquecimentos. Consequentemente, tornando-se uma opção para substituir métodos antigos de registros ainda utilizados na maioria dos locais de trabalho.

Palavras-chave: Ionic. Registro de Ponto. PostgreSQL. API. PHP.

ABSTRACT

The management of time records is essential to control the working time of a company's employees. With the use of an old system, employees end up wasting relevant time, in addition to possible forgetfulness or failures in their records. The present work consists in the development of an application using the *Ionic framework*, *PostgreSQL* database, and *API* developed in *PHP*, which is focused on the entire system records performed by company employees. As a result, the application created a more practical way to use geolocation, to check if the employee is inside their work environment to register automatically, thus solving the problem the cost of time and possible forgetfulness. Consequently, making it an option to replace old recording methods still used in most workplaces.

Palavras-chave: Ionic. Replace old methods. PostgreSQL. API. PHP.

LISTA DE ILUSTRAÇÕES

Figura 1 – Funcionamento do PHP entre cliente e servidor.	14
Figura 2 – Modelo do fluxo de dados simples PostgreSQL.	16
Figura 3 – Tela administrativa do <i>XAMPP</i>	19
Figura 4 – Tela de registro do Tangerino	20
Figura 5 – Tela do aplicativo PontoGo	21
Figura 6 – Tela de relatórios vista pelo gestor.	27
Figura 7 – Página de cadastro de novos quadros de horários.	28
Figura 8 – Página de cadastro de novos funcionários.	29
Figura 9 – Tela de <i>Login</i>	30
Figura 10 – Diagrama entidade relacionamento.	32
Figura 11 – Página de <i>login</i>	33
Figura 12 – Página principal administrativa	34
Figura 13 – Página de dados administrativa	35
Figura 14 – Página de funcionários	36
Figura 15 – Página de cadastro de funcionários	37
Figura 16 – Páginas de horários e cadastro de novo horário	38
Figura 17 – Página de relatório de todos funcionários	39
Figura 18 – Página de principal de funcionário	40
Figura 19 – Página de dados do funcionário	41
Figura 20 – Página de relatório para o funcionário	42

LISTA DE QUADROS

Quadro 1 – Comparação funcionalidades GeoPoint x Tangerino.	21
Quadro 2 – Comparação funcionalidades GeoPoint x PontoGo.	22

LISTA DE ABREVIATURAS E SIGLAS

PHP	Personal Home Page
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
API	Application Programming Interface
(Diagrama) ER	(Diagrama) Entidade Relacionamento
JSON	JavaScript Object Notation
CNPJ	Cadastro Nacional da Pessoa Jurídica
PDO	PHP Data Object
SQL	Structured Query Language
XML	eXtensible Markup Language
CPF	Cadastro de Pessoas Físicas

LISTA DE FRAGMENTOS DE CÓDIGO

Fragmento de Código 1 – Criação e um projeto Ionic utilizando linha de comando.	23
Fragmento de Código 2 – Marcação de tabelas do HTML.	23
Fragmento de Código 3 – Conexão com banco de dados utilizando PHP PDO.	24
Fragmento de Código 4 – Utilizando PHP para leitura do JSON e estruturação dos dados	25
Fragmento de Código 5 – Calculando tempo de intervalo entre dois valores	25
Fragmento de Código 6 – Leitura em tempo real das coordenadas.	29
Fragmento de Código 7 – Sucesso na tentativa de <i>login</i>	30

SUMÁRIO

1	INTRODUÇÃO	13
2	TECNOLOGIAS	14
2.1	PHP	14
2.1.1	Utilizações	14
2.2	PostgreSQL	15
2.3	Framework	16
2.3.1	Ionic	17
2.4	API	18
2.5	JSON	18
2.6	XAMPP	18
2.7	Heroku	19
2.8	Ferramentas atuais	19
2.8.1	Tangerino Controle de Ponto	20
2.8.2	PontoGo	21
2.9	Legislação	22
3	DESENVOLVIMENTO	23
3.1	Ambiente	23
3.1.1	CSS e HTML	23
3.1.2	JavaScript	24
3.2	API	24
3.3	Escopos da aplicação	25
3.3.1	Empresa	26
3.3.1.1	Administradores	26
3.3.1.2	Funcionário	26
3.3.2	Relatórios	26
3.3.3	Quadro de horários	28
3.3.4	Registro de horário ponto	29
3.4	Cadastro de novos funcionários	29
3.5	Login	30
3.6	Logout	31
3.7	Diagrama Entidade Relacionamento	31
4	RESULTADOS OBTIDOS	33
4.1	Módulos do administrador	33

4.2	Módulos do funcionário	39
5	CONCLUSÃO E TRABALHOS FUTUROS	43
	REFERÊNCIAS	45

1 INTRODUÇÃO

A realização do registro de horário ponto, relatórios e controle dos funcionários, são essenciais para uma agência ter em mãos as informações de tempo trabalhado dos seus funcionários. Partindo deste pressuposto, toda agência deve buscar um sistema que atenda suas necessidades de forma eficaz e ágil.

Com a retomada do trabalho *in loco* pós-pandemia, empresas estão buscando ferramentas para o registro de horário ponto tradicional, que efetua o registro quando o funcionário chega e sai do seu ambiente de trabalho. Como uma peça fundamental para esse propósito, a ferramenta deve sanar alguns problemas comuns, dentre eles, o esquecimento do registro. Consoante a uma matéria realizada pela Rede Jornal Contábil, as políticas da empresa devem ter legislações que atuem sobre o esquecimento do registro do horário ponto. Na maioria das legislações criadas, o funcionário que esquece de efetuar o registro é tratado da mesma forma com o que se atrasou. (JUNIOR, 2020)

Um dos métodos mais utilizados em grandes empresas é o ponto eletrônico, que inclui o registro por digital ou cartão. Suas desvantagens são o deslocamento dos funcionários até o aparelho de registro, suas falhas em casos de falta de energia ou de papel (nos que possuem a impressão do registro para o funcionário) e o custo para manutenções e instalação dos aparelhos de registro nos locais de trabalho.

Este trabalho foi pautado na intenção de sanar e agilizar a eficiência de um registro de um horário ponto da maneira com que seja tolerante a falhas humanas, criando um aplicativo que, através da utilização das coordenadas geográficas do dispositivo (*smartphone*) do usuário, é possível saber se ele está dentro da área de alcance da empresa onde trabalha, efetuando o registro ponto automaticamente. Dessa forma, resultando em uma opção que não necessita do deslocamento dos funcionários para efetuar o registro, não requer manutenções periódicas e instalações de demais aparelhos. Outro fator relevante, dentro desse contexto, é o fácil acesso às informações básicas que o funcionário ou o empregador podem necessitar durante o cotidiano de trabalho.

O trabalho divide-se em três etapas distintas. Na primeira etapa foi efetuada uma análise das aplicações mais comuns utilizadas e descartados alguns métodos antigos, como o registro por digital ou por cartão magnético. Coletando as funcionalidades essenciais e estudando as melhorias, seguiu-se para a próxima etapa. Na segunda etapa, foi realizado um estudo das ferramentas que seriam utilizadas para a implementação do aplicativo vigente, como consta no Capítulo 2. Ainda na segunda etapa, foi verificada a modelagem e a construção dos esboços das estruturas do projeto e os diagramas, com base nas informações já adquiridas. Na terceira e última etapa a aplicação foi desenvolvida e testada. Junto ao desenvolvimento, foram aprimoradas algumas funcionalidades para se adequar às normas de um registro de horário ponto. A segunda e terceira etapas são apresentadas no Capítulo 3 e no Capítulo 4.

2 TECNOLOGIAS

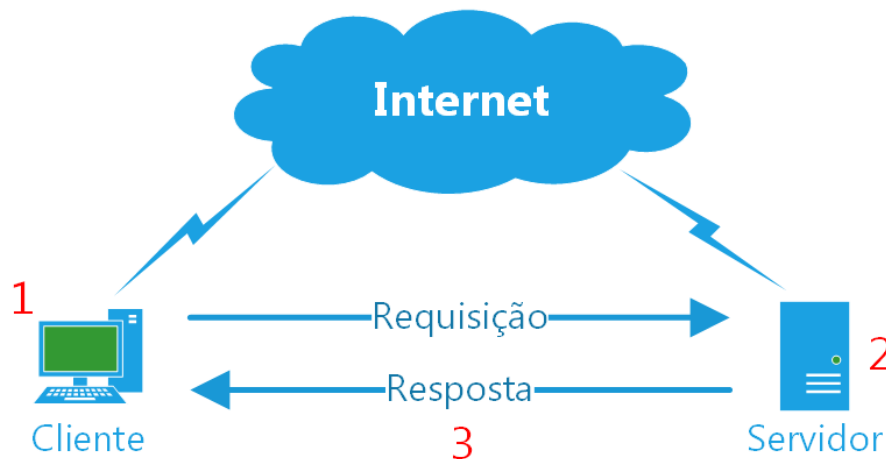
O presente trabalho foi desenvolvido utilizando tecnologias das áreas da computação que são elucidadas nas próximas seções.

2.1 PHP

O PHP *PHP:Hypertext Preprocessor* é uma linguagem que é especialmente adequada para desenvolvimento de aplicações *web*, onde pode-se fazer tudo, desde um blog até uma grande rede social. Sua grande utilização se dá pelo fato da fácil aprendizagem. Por ser uma linguagem interpretada ela não necessita de um compilador para sua execução. Dessa maneira, para utilizar um código em PHP é necessário um servidor *web* com devidas configurações. *Softwares* como o *Xampp* podem ser baixados com as configurações necessárias para a execução dessa linguagem (DEVFURIA, 2020).

O PHP em funcionamento com uma aplicação *web* pode retornar requisições *HTTP* mandadas pelo usuário, fazendo a consulta em sua base de dados ou executando uma determinada função ele irá devolver a solicitação. A Figura 1 mostra o funcionamento de um servidor *web* rodando com PHP.

Figura 1 – Funcionamento do PHP entre cliente e servidor.



Fonte: (DEVFURIA, 2020)

2.1.1 Utilizações

Devido a grande utilização do PHP, ele se tornou uma opção para aplicações com diferentes propósitos, a seguir algumas utilizações:

- *API's* - um serviço que fornecerá informações ao usuário que fizer a requisição;

- Sistemas *web* e-commerces, blogs, portais, páginas institucionais para divulgação e redes sociais;
- Rotinas - *scripts* que executam determinadas operações em um intervalo de tempo programado.

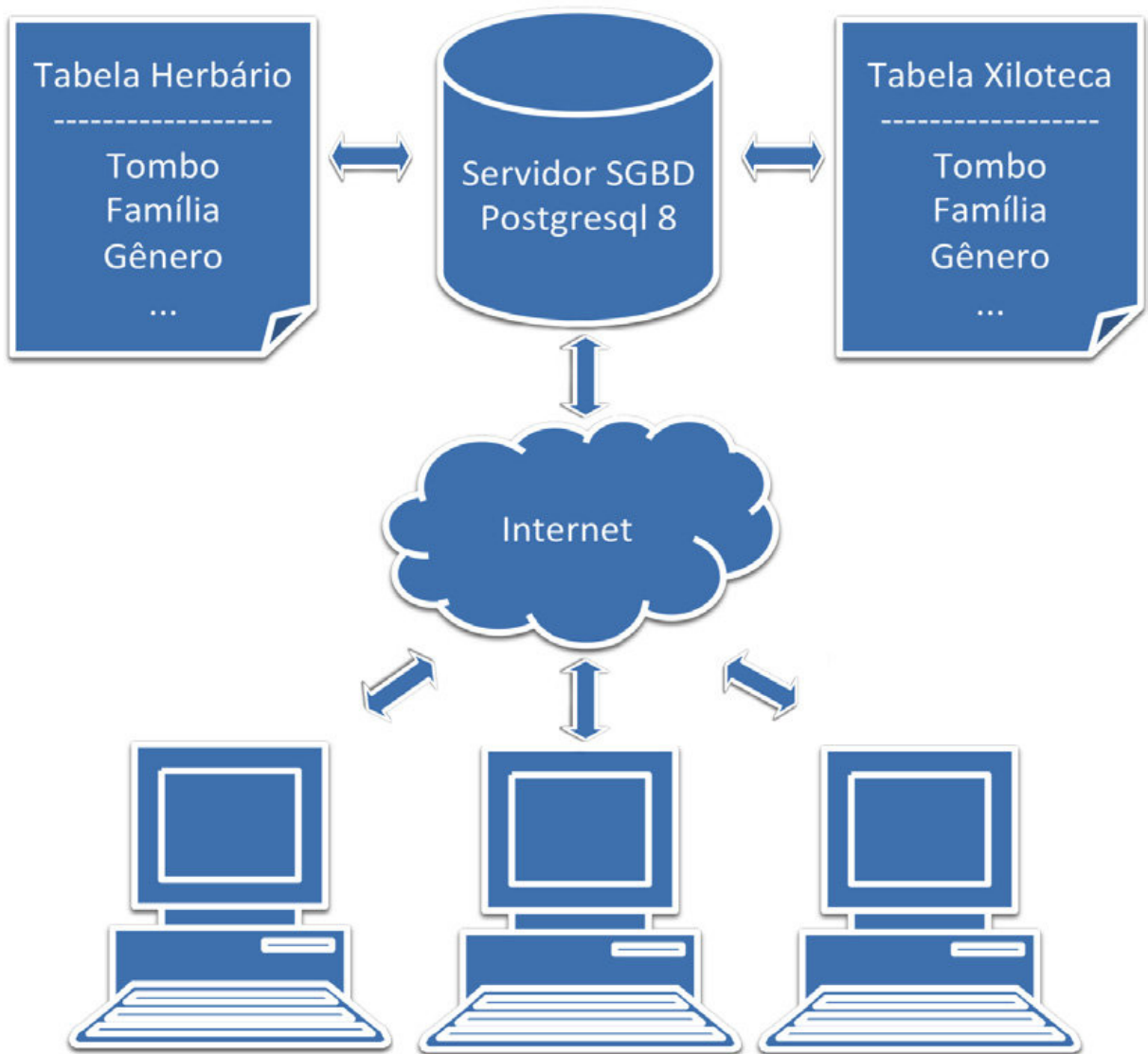
2.2 PostgreSQL

Um dos bancos de dados relacionais bastante popular é o PostgreSQL, o qual é de código aberto, muito usado para o desenvolvimento de sistemas *web*. Nele existe uma grande quantidade de ferramentas para gerenciamento e possui um bom auxílio de sua comunidade. Com mais de 30 anos de desenvolvimento ativo, esse banco de dados se tornou muito robusto e confiável (POSTGRESQL, 2021).

O projeto do POSTGRES foi liderado pelo Professor Michael Stonebraker, foi patrocinado pela Agência de Projetos de Pesquisa Avançada de Defesa (DARPA) e hoje encontra-se sobre mãos da Appendix E, empresa que continua fornecendo atualizações frequentemente para a plataforma.

As tabelas do PostgreSQL se "comunicam"entre si via *joins*, termo utilizado para definir as relações entre elas. E através das solicitações dos usuários pela rede é retornado uma determinada informação. A Figura 2 demonstra o funcionamento do lado cliente e servidor de um sistema simples que trabalha com PostgreSQL (POSTGRESQL, 2021).

Figura 2 – Modelo do fluxo de dados simples PostgreSQL.



Fonte: (POSTGRESQL, 2021)

O PostgreSQL possui fatores que o tornam uma ótima opção para o desenvolvimento de aplicações *web*:

- Fácil integração com aplicações em PHP;
- Pode ser executado em um computador pessoal;
- Não precisa de uma máquina com alta performance para uma execução rápida.

2.3 Framework

Existem muitas opções disponíveis para o desenvolvimento de aplicativos para dispositivos móveis. A tendência atual é a utilização de *framework* híbrido, onde não é necessário criar o aplicativo usando a linguagem nativa da plataforma, nesse desenvolvimento, é possível usar recursos mais conhecidas como *HTML*, *CSS* e *JavaScript*, sendo assim um projeto

mais flexível, com interfaces mais diversificadas e novos recursos tecnológicos (ZANETTI, 2020).

O *framework* pode ser entendido como uma base que contém muitas funções, onde o desenvolvedor pode utilizá-las ou não de acordo com a necessidade, agilizando o processo de desenvolvimento.

A utilização desse tipo de ferramenta já é muito comum entre os desenvolvedores de aplicativos. Com a finalidade de auxiliar no método ágil de desenvolvimento, reduzindo custos e aumentando a produtividade, muitos *frameworks* vêm surgindo e sendo atualizado corriqueiramente.

Uma das principais vantagens ao se utilizar um *framework* em algum projeto, é a dispensabilidade na reescrita de códigos, o que permite focar somente na resolução dos problemas, ou seja, tudo pensado na implementação é com o intuito de chegar ao objetivo final (ZANETTI, 2020).

- Menor quantidade de erros - código já testado anteriormente e aprimorado faz com que com o passar do tempo a quantidade de erros se torne cada vez mais rara;
- Redução de custos - a redução de custo em um desenvolvimento de software está ligada diretamente ao tempo de desenvolvimento. Evitando reescrita de funções, a equipe de desenvolvimento estará destinada a atender a demanda final solicitada.
- Fácil aprendizagem - a grande comunidade de pessoas utilizando o mesmo *framework* faz com que seja comum alguém já ter executado a demanda que você precisa, compartilhando a solução ou a melhor prática para o desenvolvimento dela, além disso a documentação é extremamente minuciosa e extensa, com uma ampla gama de conteúdos;
- Padronização de código - auxiliando na compartilhabilidade do sistema, o *framework* e suas funções específicas, faz com que o desenvolvedor siga esse padrão para implementar o sistema.

2.3.1 Ionic

Conforme citado anteriormente, diversos *frameworks* estão disponíveis para o desenvolvimento de aplicativos. No presente trabalho foi utilizado o *Ionic* como código-fonte base.

O *Ionic* é um *framework* de código aberto muito famoso desenvolvido com a tecnologia *PhoneGap* da Adobe (ZANETTI, 2020).

Este *framework* tem como sua linguagem base o *JavaScript*, que é responsável pela construção da camada lógica do desenvolvimento e interação, o *HTML* é empregado na estruturação do conteúdo que o usuário verá e por fim o *CSS* que é utilizado na formatação e estilização da aplicação. Uma tecnologia embutida no *framework* é o *AngularJS*, plataforma do Google para o desenvolvimento *front-end* em aplicações para *web* (ZANETTI, 2020).

Algumas vantagens de se utilizar o *Ionic* estão listadas a seguir:

- Estilização - o *framework* já vem com uma grande quantidade de ferramentas de estilos

que seguem a melhor usabilidade do usuário, auxiliando o desenvolvedor a criar uma aplicação intuitiva;

- Documentação - a documentação do *Ionic* é muito explicativa, em que é possível encontrar rapidamente tudo que se precisa saber sobre o *framework*;
- *Ionic serve* - o seguinte comando é utilizado para fazer a aplicação ser exibida previamente em um navegador, simulando um celular, fazendo com que o desenvolvedor possa rapidamente codificar e visualizar o resultado;
- Exportação - o *Ionic* possui comandos para a exportação tanto para *Android* quanto para *IOS* facilitando a geração do arquivo de instalação para cada dispositivo.

2.4 API

API, Application Programming Interface ou em português, Interface de Programação de Aplicativos. *API* é um conjunto de padrões que fazem parte de uma interface, permitindo a criação de outras plataformas de maneira mais simples para os desenvolvedores (MEDEIROS, 2014).

Aplicativos de celulares que precisam fazer consulta a informação normalmente irão fazer a requisição a uma *API* já desenvolvida, dessa maneira também é feita a abstração das funcionalidades do aplicativo.

Na utilização de uma *API* não é necessário que o desenvolvedor saiba tudo sobre a mesma, basta ter acesso à documentação para que consiga fazer a busca das instruções necessárias para a execução da função (MEDEIROS, 2014).

2.5 JSON

JSON, JavaScript Object Notation é uma tecnologia que teve origem em 2002 e pode ser comparado ao XML. *JSON* é um formato de dados leve para a comunicação entre sistemas, ele suporta objetos e tem uma ótima velocidade no transporte e execução de dados, por isso é muito utilizado em aplicativos, por estarem em constante comunicação com o sistema que disponibiliza as informações (GAME, 2011).

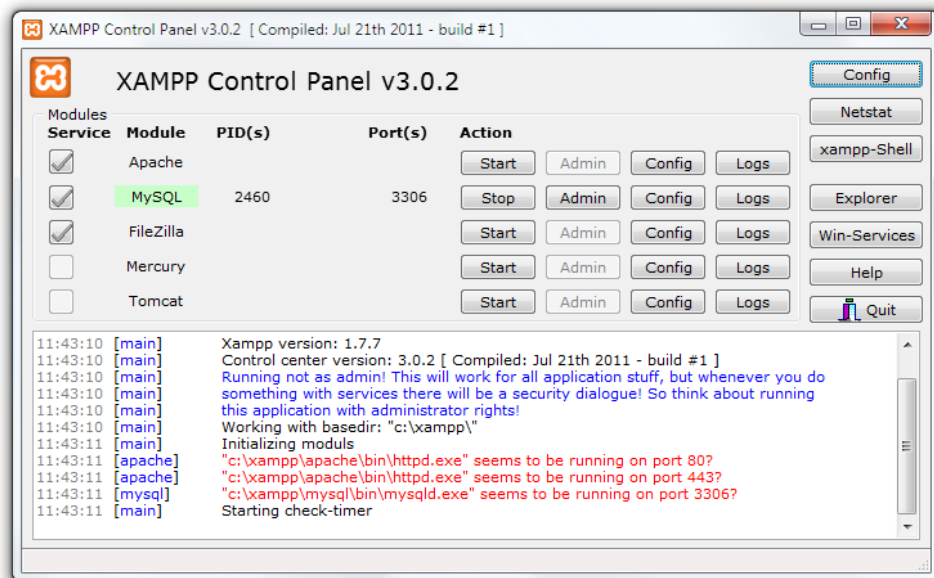
2.6 XAMPP

XAMPP é um conjunto de softwares que instala este pacote de programas de forma única. Sua sigla é uma abreviação para *Apache, MySQL, PHP e Pearl* sendo o X inicial para identificar que esta ferramenta é multiplataforma. Com ele vem as principais ferramentas para um desenvolvimento *web*, entre elas o banco de dados para o armazenamento das informações e o *Apache* com suporte para as linguagens *PHP e Perl* (HIGA, 2012).

O *software* funciona como um servidor virtual dentro da máquina do desenvolvedor, assim é possível realizar uma série de ajustes para que o ambiente de trabalho transforme-se em

algo descomplicado. O painel de gerenciamento da ferramenta é muito intuitivo, como mostra a Figura 3.

Figura 3 – Tela administrativa do XAMPP



Fonte: (HIGA, 2021)

2.7 Heroku

HEROKU é uma plataforma de hospedagem muito utilizada para testes em produção, por ser integrada com o *GITHUB*, plataforma de versionamento, facilitando o *deploy* das aplicações. A grande quantidade de serviços de hospedagens disponibilizadas pelo *Heroku* a faz com que seja muito usada, além de possuir uma versão gratuita, que acaba atraindo pequenos projetos (PACHECO, 2020).

Para aplicações gratuitas existem algumas desvantagens, pois a utilização de um pequeno servidor faz com que ele seja gerenciado de uma maneira com que em 30 minutos sem a utilização ele entre em modo adormecido. Quando um usuário fizer uma requisição ele irá levar um tempo a mais para que volte ao funcionamento normal (PACHECO, 2020),

2.8 Ferramentas atuais

Para realizar o registro de horário ponto de agências, os gestores possuem uma grande quantidade de ferramentas disponíveis no mercado. Muitas delas usam tecnologias antigas e algumas estão surgindo com tecnologias mais recentes e eficazes. O presente aplicativo foi desenvolvido com base em alguns módulos que esses sistemas mais recentes carregam, simplificando algumas das principais funcionalidades com a finalidade de tornar mais prático e

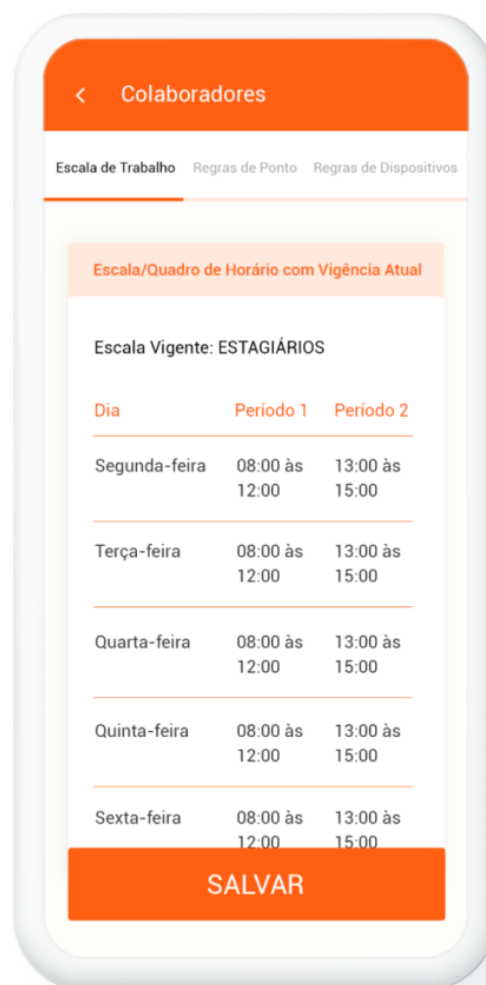
rápido o registro do horário ponto. A seguir são listadas as plataformas usadas como base para o desenvolvimento do sistema.

2.8.1 Tangerino Controle de Ponto

O Tangerino é uma ferramenta que dispõem de uma série de fatores positivos para o gerenciamento do registro. Por possuir um registro de ponto por senha, foto e localização, ele tem uma enorme segurança. O *layout* do Tangerino é bastante simples e de fácil acesso.

Um aspecto muito importante de um registro de horário ponto é o relatório de registros, onde o tangerino se mostra muito eficaz, principalmente para o usuário final, como mostra a Figura 4.

Figura 4 – Tela de registro do Tangerino



O Quadro 1 mostra uma comparação de funcionalidades do sistema desenvolvido e do Tangerino.

Quadro 1 – Comparação funcionalidades GeoPoint x Tangerino.

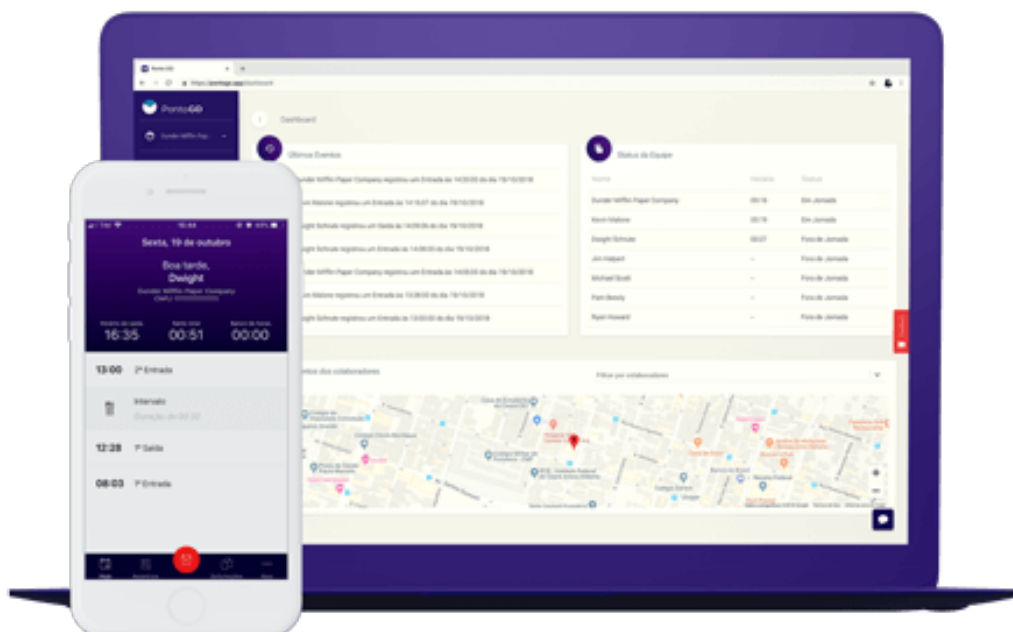
	GeoPoint	Tangerino
Relatórios	X	X
Cadastro de Funcionários	X	X
Agilidade	X	-
Administrativo	X	X
Carga Horária	X	X
Segurança	X	X

2.8.2 PontoGo

Outra aplicação disponível no mercado para o registro de horário ponto de uma agência, é o PontoGo. Essa plataforma conta com um *layout* muito mais agradável e intuitivo, tornando ainda melhor a usabilidade, para o registro ainda é necessário que o usuário faça uma interação com o aplicativo pressionando no botão para salvar o ponto.

Nele é possível ver aonde a pessoa estava quando fez o registro e notifica os gestores. Na figura Figura 5 é possível ver a tela do aplicativo PontoGo.

Figura 5 – Tela do aplicativo PontoGo



O Quadro 2 elenca as funcionalidades do sistema desenvolvido e do PontoGo.

Quadro 2 – Comparação funcionalidades GeoPoint x PontoGo.

	GeoPoint	PontoGo
Relatórios	X	X
Cadastro de Funcionários	X	X
Agilidade	X	-
Acesso administrativo	X	X
Carga Horária	X	X
Segurança	X	X
Privacidade	X	-
Notificação	-	X

2.9 Legislação

Sabendo que o registro de ponto para estabelecimentos com mais de 20 trabalhadores é considerado obrigatório, foi necessário buscar informações sobre as normas as quais o aplicativo deve seguir. No ano de 1989 foi permitido que o registro de pontos passasse a ser feito através de meios manuais e eletrônicos e com o avanço da tecnologia os pontos eletrônicos se tornaram a maioria em relação ao registro manual.

A portaria 373, opção complementar à portaria 1510, também conhecida como lei do ponto eletrônico, menciona que os sistemas de pontos eletrônicos não podem impor restrições à marcação do registro ponto mesmo que esteja fora das normas trabalhistas, com isso foi permitido dentro do aplicativo em questão que o funcionário efetuasse os registros e o setor administrativo visualizasse possíveis descumprimentos com a lei e conseqüentemente tomar devidas providencias (VASCONCELOS, 2021),

3 DESENVOLVIMENTO

Neste capítulo são aprofundadas as ferramentas utilizadas no desenvolvimento do aplicativo, assim como as linguagens, os módulos implementados e suas respectivas funcionalidades.

3.1 Ambiente

A primeira etapa para iniciar o desenvolvimento do aplicativo foi a escolha da linguagem e do *framework* que se encaixaria melhor às necessidades. O *framework* escolhido foi o *Ionic*, por ser de fácil aprendizado para o desenvolvimento e atender de maneira satisfatória as questões de usabilidade do consumidor final, juntamente com muitas funcionalidades embarcadas que foram utilizadas no trabalho. Com a escolha do *Ionic* (explanado na Subseção 2.3.1) obteve-se como consequência a necessidade de utilizar *AngularJS* devido a sua construção. Nela foi utilizado *CSS* para a estilização do projeto, *HTML* para a estruturação das telas e *JavaScript* como linguagem principal para as funcionalidades.

Fragmento de Código 1 – Criação de um projeto Ionic utilizando linha de comando.

```
1 ionic start <name> <template> [options]
```

O comando citado no Fragmento de Código 1 faz a criação de toda a estrutura de pastas e arquivos de um projeto base utilizando *Ionic*. Esse procedimento otimiza o tempo de desenvolvimento, agilizando a implementação da parte lógica do aplicativo.

3.1.1 CSS e HTML

Para este trabalho foi utilizado o *CSS* para a estilização e *HTML* para a estruturação. Nas telas que possuíam formulários seguiu-se um padrão utilizando boas práticas para que os usuários saibam o que deve ser preenchido em cada campo. Os botões foram feitos tendo como base alguns modelos que são disponibilizados pelo *Ionic*.

A criação das tabelas de horários foi uma etapa importante onde utilizou-se a marcação de tabelas do *HTML* para melhor apresentação, para que os dados não ficassem conflitantes visualmente para o usuário, como pode ser visto no Fragmento de Código 2.

Fragmento de Código 2 – Marcação de tabelas do *HTML*.

```
1 <table>
2   <tbody>
3     <tr>
4       <td>1</td>
```

```

5     <td>2</td>
6   </tr>
7   <tr>
8     <td>1</td>
9     <td>2</td>
10  </tr>
11 </tbody>
12 </table>

```

3.1.2 JavaScript

O *JavaScript* presente na programação é utilizado para adicionar interatividade. Em aplicativos, sua utilização é fundamental, exemplificando a atividade de um usuário efetuar o acesso ao sistema, onde o mesmo digita o seu email, senha e clica no botão "Acessar". Nesse momento o *JavaScript* capta esses dados e envia para a *API* informando que é uma ação para o acesso do usuário. A *API* pode retornar uma resposta verdadeira ou falsa. Com isso, o *JavaScript* irá reagir mostrando na tela um sucesso ou avisá-lo de que houve um problema.

3.2 API

O aplicativo precisa obrigatoriamente ter uma conexão com um banco de dados onde estão as informações para o seu funcionamento. Essa conexão é feita através de uma *API* escrita com a linguagem de código *PHP* citado no Capítulo 2.

No desenvolvimento foi feita a conexão com o banco de dados *PostgreSQL* com a função *PDO*, como mostra o Fragmento de Código 3.

Fragmento de Código 3 – Conexão com banco de dados utilizando PHP PDO.

```

1 <?php
2 $conn = new PDO("pgsql:host=localhost;port=;dbname=geopoint;
   user=root;password=rootpass");
3 ?>

```

Com a conexão estabelecida é feito o desenvolvimento das funções que irão buscar, salvar e atualizar os dados de acordo com as requisições vindas dos usuários do aplicativo. A *API* desenvolvida para o trabalho não se estende muito além das necessidades básicas. Com a hospedagem desse código junto ao banco de dados no *Heroku* (citado no Capítulo 2), é possível iniciar o desenvolvimento das requisições do aplicativo.

As requisições feitas pelo *Ionic* são no formato de *JSON*, citada no Capítulo 2. Dessa maneira, são utilizadas as funções do *PHP*, fazendo a leitura e estruturação dos dados que irão ser utilizados nos laços de repetição para inserção de uma única vez, agilizando todos os processos executados pelo aplicativo. É possível ver como é feita a interpretação do *JSON* no Fragmento de Código 4.

Fragmento de Código 4 – Utilizando PHP para leitura do JSON e estruturação dos dados

```

1 <?php
2     $postjson = json_decode(file_get_contents('php://input'),
3         true);
4     $result = json_encode(array('success'=>true, 'result'=>
5         $dados));
6 ?>

```

A etapa de grande relevância no desenvolvimento da *API* foi o cálculo do tempo que o usuário esteve na empresa, fazendo o registro com o horário base do servidor, código do usuário e qual foi a ação, diferenciada entre entrada e saída. Após isso, o aplicativo realiza a solicitação de um novo registro somente quando o usuário mudar o estado anterior. No caso do usuário estar na empresa, a nova solicitação será realizada somente quando o usuário sair do alcance geográfico da empresa. Com os dois registros é possível calcular o tempo dentro do local de trabalho. O mesmo se repetirá todos os momentos que o usuário executar essa atividade. O Fragmento de Código 5 mostra como é calculada a diferença entre duas datas utilizando a linguagem de programação em *PHP*.

Fragmento de Código 5 – Calculando tempo de intervalo entre dois valores

```

1 <?php
2     $d1 = new DateTime("2021-06-10 08:02:20");
3     $d2 = new DateTime("2021-06-10 11:55:45");
4     $interval = $d1->diff($d2);
5 ?>

```

3.3 Escopos da aplicação

O primeiro passo tomado para o desenvolvimento do trabalho foi o mais relevante, constituído através da definição dos escopos utilizados pelos usuários que irão acessar o aplicativo. Para o controle diretamente pelos gestores da empresa, foi necessário atribuir funções onde o usuário administrativo terá acesso e o usuário funcionário não terá. Além disso, é necessário que o usuário funcionário possa ter sua determinada carga horária vinculada.

A melhor tratativa para uma situação de restrições, onde existe apenas dois níveis de usuário, pode simplesmente ser feita diretamente no código, verificando qual o tipo de usuário está requisitando o acesso ao módulo, sem a necessidade de uma tabela auxiliar para a atribuição de permissões e níveis de usuários.

3.3.1 Empresa

A empresa é um ponto que contém informação crucial para o desenvolvimento do aplicativo. Nela temos a informação dos pontos de latitude e longitude da empresa, que é indispensável para o registro do horário ponto através da localização do funcionário.

Todos usuários estão obrigatoriamente vinculados a uma empresa, tanto como administrador ou como funcionário. Ela também possui algumas informações básicas como CNPJ e nome.

Os tipos de usuários de uma empresa são divididos em duas funções: administrador e funcionário. Os mesmos serão apresentados nas próximas subseções.

3.3.1.1 Administradores

Usuário administrador é o responsável pela empresa, tendo todas as permissões existentes no aplicativo, tais como:

- Criar, editar e excluir funcionários;
- Criar, editar e excluir quadro de horários;
- Editar dados da empresa;
- Editar registro de pontos do funcionário sob demanda;
- Editar dados do próprio usuário administrativo;
- Visualizar relatórios de todos funcionários da empresa que administra.

A tela inicial do usuário administrador também é diferente, para que não apareçam informações sem necessidade ao usuário designado como funcionário.

3.3.1.2 Funcionário

O funcionário de uma empresa é o usuário que possui poucas permissões devido à simplicidade da ferramenta para torná-la mais ágil. Esse nível de usuário pode ver alguns relatórios para controle próprio e realizar as seguintes ações:

- Editar os dados do próprio usuário;
- Criar registro de ponto;
- Visualizar relatório do próprio usuário.

3.3.2 Relatórios

O módulo de relatórios é indispensável para qualquer aplicativo de horário ponto. Com ele podem ser definidas quantas horas os funcionários estiveram dentro da empresa exercendo

suas funções. Esse relatório consiste no agrupamento dos registros de horário ponto para calcular, juntamente com o quadro de horários que será apresentado na próxima sessão. Com a união do quadro de horários e os registros é possível calcular, por exemplo, quantas horas a mais o funcionário fez por dia ou quantas horas ele fez a menos.

O usuário administrativo pode tanto visualizar todos os relatórios de sua empresa, como editar os registros para solucionar possíveis esquecimentos de um funcionário ou para casos de atestados médicos e afins, como mostra a Figura 6.

Figura 6 – Tela de relatórios vista pelo gestor.

Murilo		
Sep 2, 2021		
Sep 2, 2021		
📅 Quinta	02/09	+08:36
		↩️ 08:03
		👉 11:55
		↩️ 13:15
		👉 17:59
Entrada	Saída	📌

Fonte: (AUTOR, 2021)

Além disso, é possível filtrar por datas para visualizar um relatório mensal ou semanal dos registros. No final do relatório é apresentado um somatório das horas trabalhadas dentro do escopo selecionado. Para funcionários, algumas informações devem ser restringidas, exibindo na tela apenas uma quantidade de informações focada nas necessidades do utilizador.

3.3.3 Quadro de horários

O quadro de horários tem como funcionalidade definir qual é o horário de entrada e de saída do funcionário. Com ele, é possível saber a carga horária semanal do mesmo. O quadro de horários pode ser diferente para cada funcionário e devem ser listados somente os que foram cadastrados pela empresa vigente. O administrador da empresa deve selecionar qual é o quadro do funcionário na hora do cadastro. Para o quadro de horários são necessárias algumas informações simples como mostra a Figura 7.

Figura 7 – Página de cadastro de novos quadros de horários.

VOLTAR ↩

48 Horas semanais

Segunda-Feira

08:00

10:30

14:00

16:00

Terça-Feira

08:00

Fonte: (AUTOR, 2021)

O aplicativo desenvolvido nesse trabalho foi projetado para suportar dois turnos de trabalho por dia. Por isso, é limitado ao cadastro de dois horários de entrada e dois horários de saída.

3.3.4 Registro de horário ponto

O *Ionic* possui uma funcionalidade nativa para a utilização de geolocalização, onde o usuário permite a utilização dos dados para obter latitude e longitude. O Fragmento de Código 6 tem como objetivo verificar as coordenadas do usuário.

Fragmento de Código 6 – Leitura em tempo real das coordenadas.

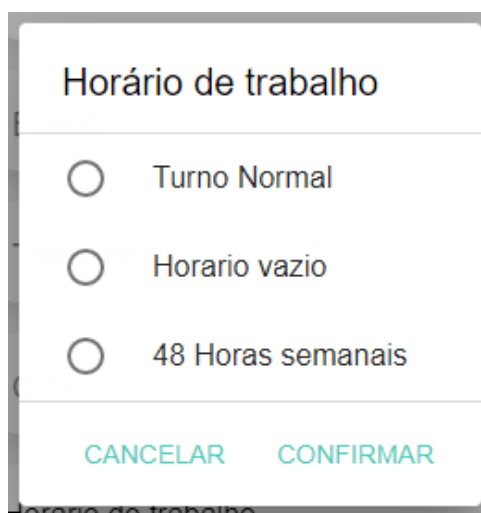
```
1 let watch = this.geolocation.watchPosition();  
2 watch.subscribe((data) => {  
3 });
```

O funcionário, ao efetuar *login* no aplicativo, faz uma requisição para a *API* solicitando as informações da empresa que pertence, salvando-as no celular para que não seja necessário estar a todo instante refazendo a requisição. Com as coordenadas da empresa, basta verificar se o funcionário está dentro do alcance e emitir para a *API* uma nova requisição, onde é efetuado o registro com o horário, código do funcionário e a chave da atividade que indicará se ele saiu ou entrou na empresa. A verificação é feita pela *API* confirmando que não existem dois registros com a mesma atividade, para que não ocorram falhas.

3.4 Cadastro de novos funcionários

Este módulo define a ocorrência de novos funcionários cadastrados no aplicativo. Para o trabalho vigente foi desenvolvida uma página para criação de um novo funcionário, contendo as informações relevantes para o funcionamento dos demais módulos. Um dos itens principais é mostrado na Figura 8.

Figura 8 – Página de cadastro de novos funcionários.



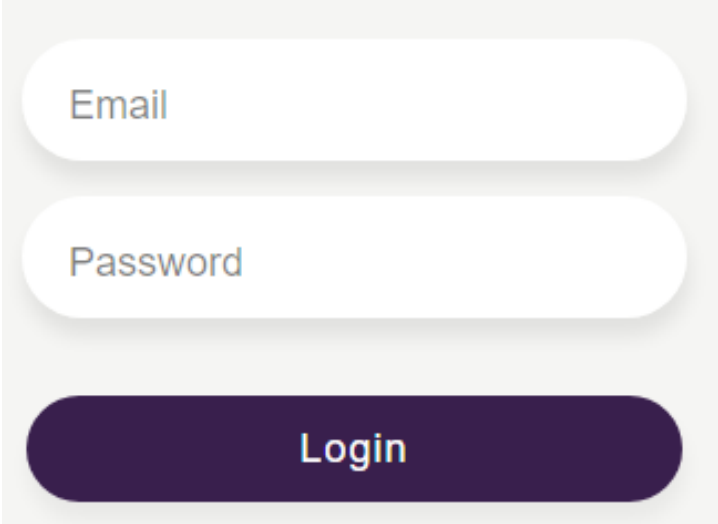
Fonte: (AUTOR, 2021)

Nessa sessão o gestor deve preencher todos os dados e selecionar o quadro de horários em que o funcionário se enquadra. Essa seleção afeta o módulo de relatórios, pois será efetuado o cálculo das horas devedoras ou sobressalentes com base nos horários de entradas e saídas. O funcionário por sua vez está ligado a todas as atividades citadas anteriormente.

3.5 Login

Um dos itens importantes para o funcionamento correto do aplicativo é a seção de *login*, onde são definidas todas as funcionalidades que o usuário tem acesso e todas as atividades que ele pode executar. Capturando as informações necessárias para verificar se o usuário existe na base de dados e se a senha está correta, foi desenvolvido uma simples tela com um formulário, como é possível ver na Figura 9.

Figura 9 – Tela de *Login*.

A imagem mostra uma interface de usuário para login. Ela contém dois campos de entrada de texto, um para 'Email' e um para 'Password', ambos com bordas arredondadas e um efeito de sombra. Abaixo dos campos, há um botão de login com o texto 'Login' em branco sobre um fundo escuro, também com bordas arredondadas e um efeito de sombra.

Fonte: (AUTOR, 2021)

Para a autenticação do usuário é utilizado o email como base. Após isso, é utilizado o algoritmo de *hashing* MD5 para a criptografia da senha enviada ser de acordo com a salva no banco no momento do cadastro do usuário. Caso as informações não estejam corretas, é mostrada na tela a mensagem de senha incorreta. Caso as informações estejam corretas, a *API* retorna, junto com a informação de sucesso, as informações da empresa a qual o usuário está atrelado e o seu nível de acesso entre funcionário e administrador. O Fragmento de Código 7 mostra como a etapa no caso de sucesso ao login se torna simples com a utilização das ferramentas do *Ionic*.

Fragmento de Código 7 – Sucesso na tentativa de *login*

```
1   if(data['success']){  
2     this.storage.set('session_storage', data['result']);
```

```
3     if(data['result']['nivel'] == 'Administrador'){
4         this.router.navigate(['/home']);
5     }else{
6         this.router.navigate(['/emphome']);
7     }
8
9     this.storage.set('session_storage_ponto', false);
10
11     const toast = await this.toast.create({
12         message: 'Logado com Sucesso!',
13         duration: 1000,
14         color: 'success'
15     });
16     toast.present();
17     this.email = "";
18     this.senha = "";
19 }else{
20     const toast = await this.toast.create({
21         message: alert,
22         duration: 1000,
23         color: 'danger'
24     });
25     toast.present();
26 }
```

3.6 Logout

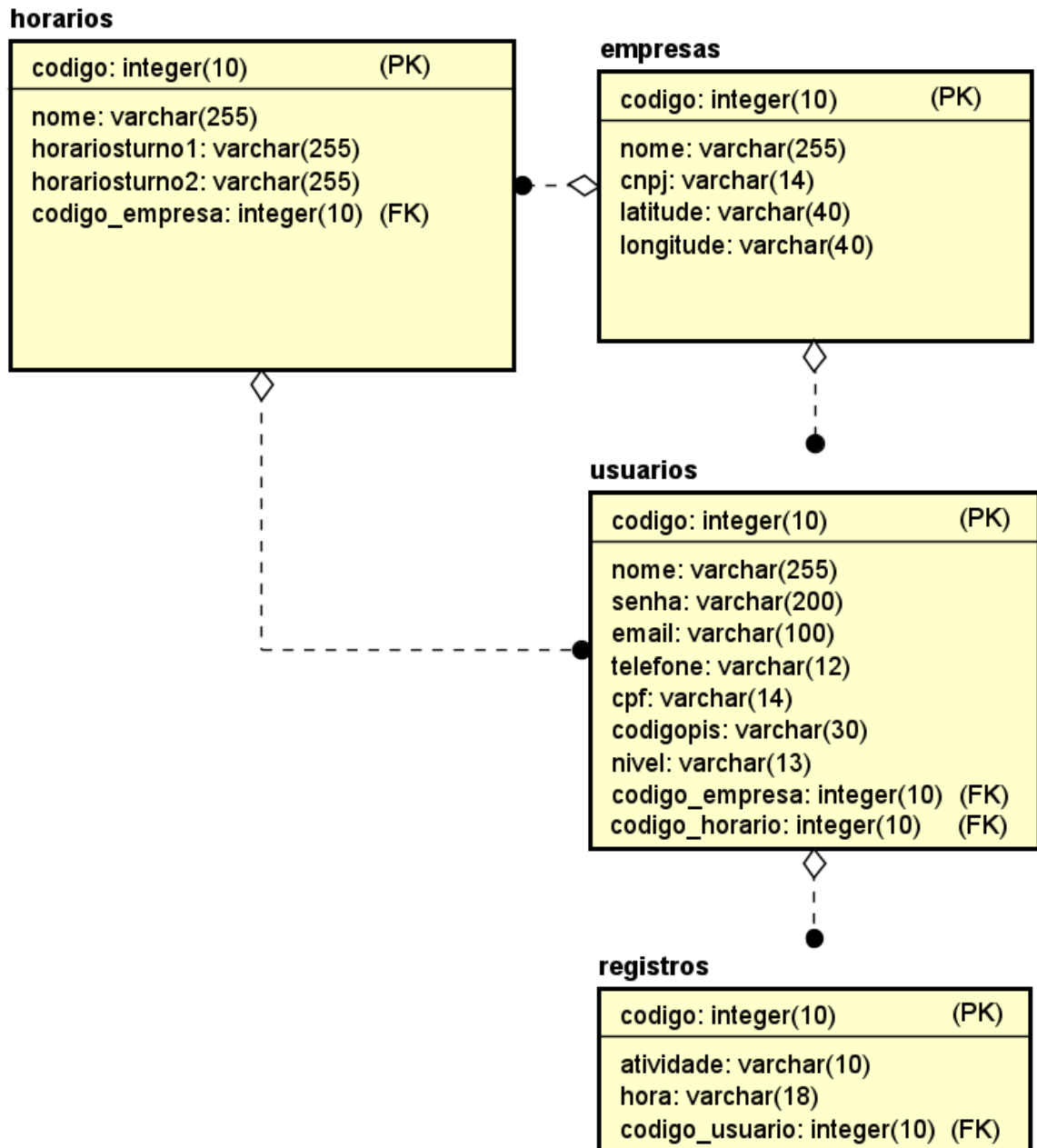
Assim como a funcionalidade para o usuário efetuar o acesso ao sistema, é necessário também que o mesmo possa sair dele, voltando para a tela de acesso. No canto superior direito da tela principal encontra-se um botão com um ícone que representa a saída por uma porta seguido da palavra "Sair". Ao clicá-lo, o usuário encerra o acesso do sistema. Caso ele volte a fazer o acesso, todo o processo será efetuado novamente.

3.7 Diagrama Entidade Relacionamento

Essa seção mostra, através da Figura 10, o Diagrama Entidade Relacionamento dos módulos que compõem o aplicativo. O diagrama é constituído pelos registros de horários, empresas, usuários e quadro de horários. É possível ver quais são os dados utilizados para o

funcionamento de cada módulo e as chaves principais para a relação entre eles.

Figura 10 – Diagrama entidade relacionamento.



Fonte: (AUTOR, 2021)

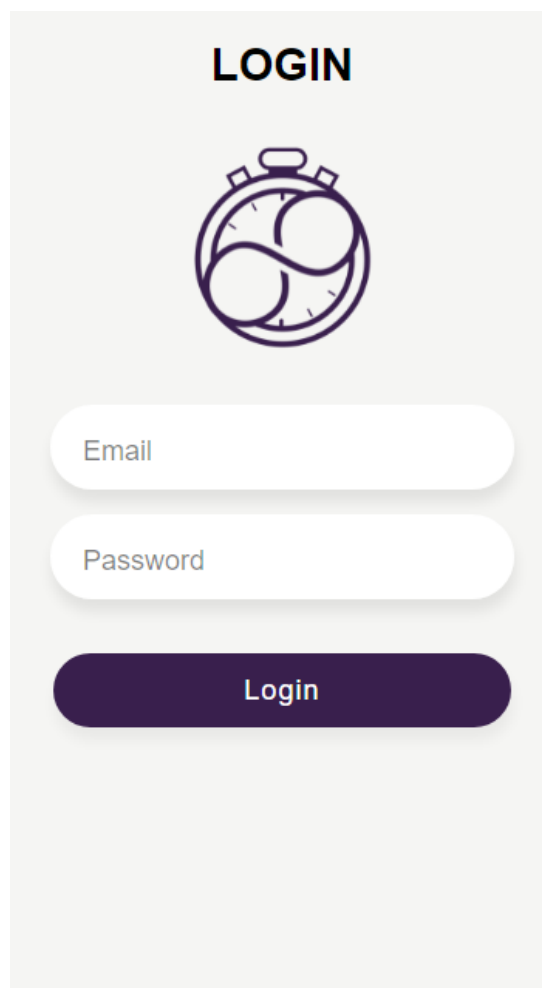
4 RESULTADOS OBTIDOS

Este capítulo tem como intuito demonstrar os resultados obtidos com o desenvolvimento do aplicativo. Nesse sentido, uma série de imagens das telas do aplicativo são distribuídas de forma organizada no decorrer dessa segmentação. A distribuição dos *prints* é dada por função do aplicativo, com cada módulo exposto separadamente.

4.1 Módulos do administrador

O módulo de funcionamento para usuários administrativos funciona de uma maneira diferente dos demais usuários, iniciando com a página de *login*. Esta página faz a solicitação do email e senha do usuário e realiza o redirecionamento para a tela principal, caso o as credenciais estiverem corretas. A tela descrita pode ser vista na Figura 11.

Figura 11 – Página de *login*



Fonte: (AUTOR, 2021)

Após a conclusão bem sucedida da etapa de *login*, a tela principal de acesso é mostrada e o administrador poderá ser direcionado para a atividade que deseja executar, clicando nas opções visíveis. A tela descrita pode ser vista na Figura 12.

Figura 12 – Página principal administrativa



Fonte: (AUTOR, 2021)

A primeira opção que o usuário pode selecionar, mostrada na Figura 12, é a opção "MEUS DADOS". Nesta, o administrador pode atualizar os dados próprios e também os dados da empresa. Com as informações básicas de nome, email, telefone e CPF para o usuário e Nome, CNPJ, latitude e longitude para a empresa. A tela pode ser vista na Figura 13.

Figura 13 – Página de dados administrativa

VOLTAR ↶

Meus Dados

Murilo

murilo@geo.com

54999854297

023.786.540-81

Dados da Empresa

GeoPoint

27.398.279/0001-60

-27.626883

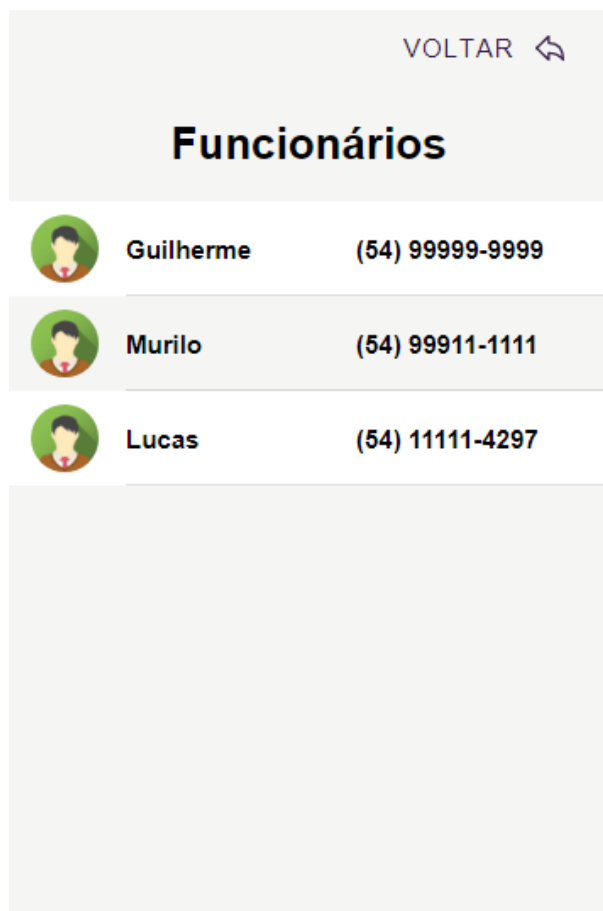
-52.277300

Salvar

Fonte: (AUTOR, 2021)

A segunda opção que o usuário pode selecionar, mostrada na Figura 12, é a opção "FUNCIONÁRIOS". Nesta tela, o administrador pode visualizar quais são todos os funcionários registrados na empresa a qual ele administra. Com nome e telefone, é possível contactar o funcionário. A tela pode ser vista na Figura 14.

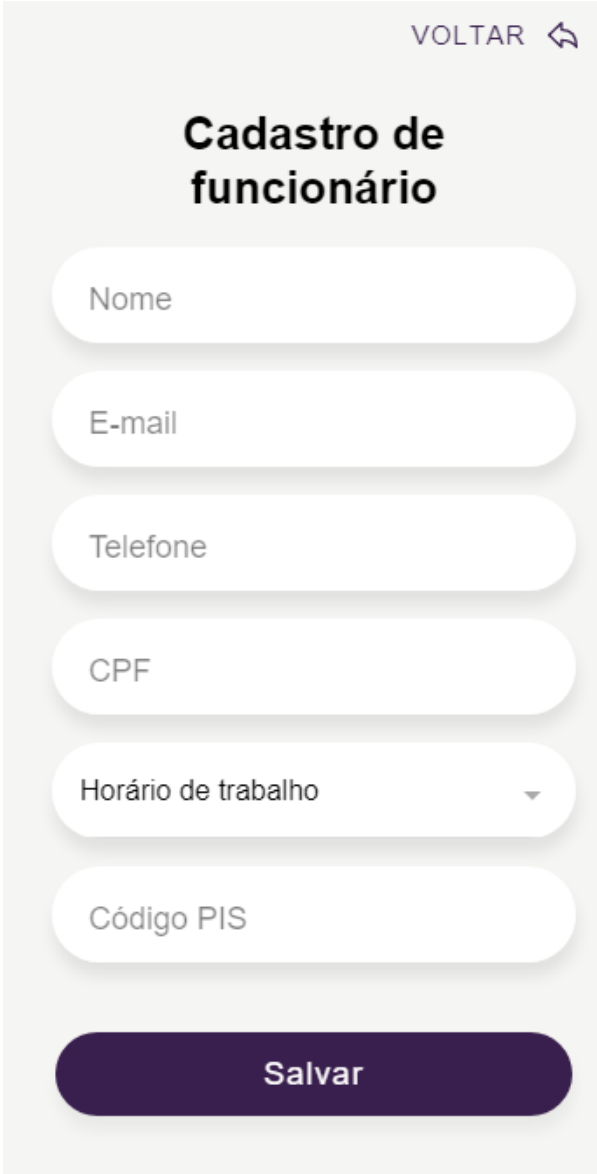
Figura 14 – Página de funcionários



Fonte: (AUTOR, 2021)

Na terceira opção, mostrada na Figura 12, é a opção "CADASTRAR", onde o administrador pode fazer o registro de um novo funcionário para a utilização do sistema, preenchendo os dados básicos, selecionando o horário de trabalho que o funcionário irá exercer e o código PIS para fins de identificação legal. A tela pode ser vista na Figura 15.

Figura 15 – Página de cadastro de funcionários



VOLTAR ↶

Cadastro de funcionário

Nome

E-mail

Telefone

CPF

Horário de trabalho ▼

Código PIS

Salvar

Fonte: (AUTOR, 2021)

A quarta opção do usuário administrativo, mostrada na Figura 12, é a opção "HORÁRIOS". Nesta etapa, o usuário administrativo pode visualizar a lista de horários que foram cadastrados para a empresa que administra e editá-los. A mesma tela conta com um botão "Adicionar", para que o usuário possa fazer um registro de um novo horário. Os horários devem sempre conter a hora da entrada e saída de cada turno, podendo ser apenas 1 turno por dia, dois turnos ou nenhum preenchimento, caso o funcionário não tenha expediente de trabalho no dia da semana em questão. As telas descritas podem ser vistas na Figura 16.

Figura 16 – Páginas de horários e cadastro de novo horário

VOLTAR ↩

Tabela de Horários

Turno Normal	EDITAR ✎
Horario vazio	EDITAR ✎
48 Horas semanais	EDITAR ✎

Adicionar

Nome do horário

Segunda-Feira

Horário de Entrada (Manhã)

Horário de saída (Manhã)

Horário de Entrada (Tarde)

Horário de saída (Tarde)

Terça-Feira

Horário de Entrada (Manhã)

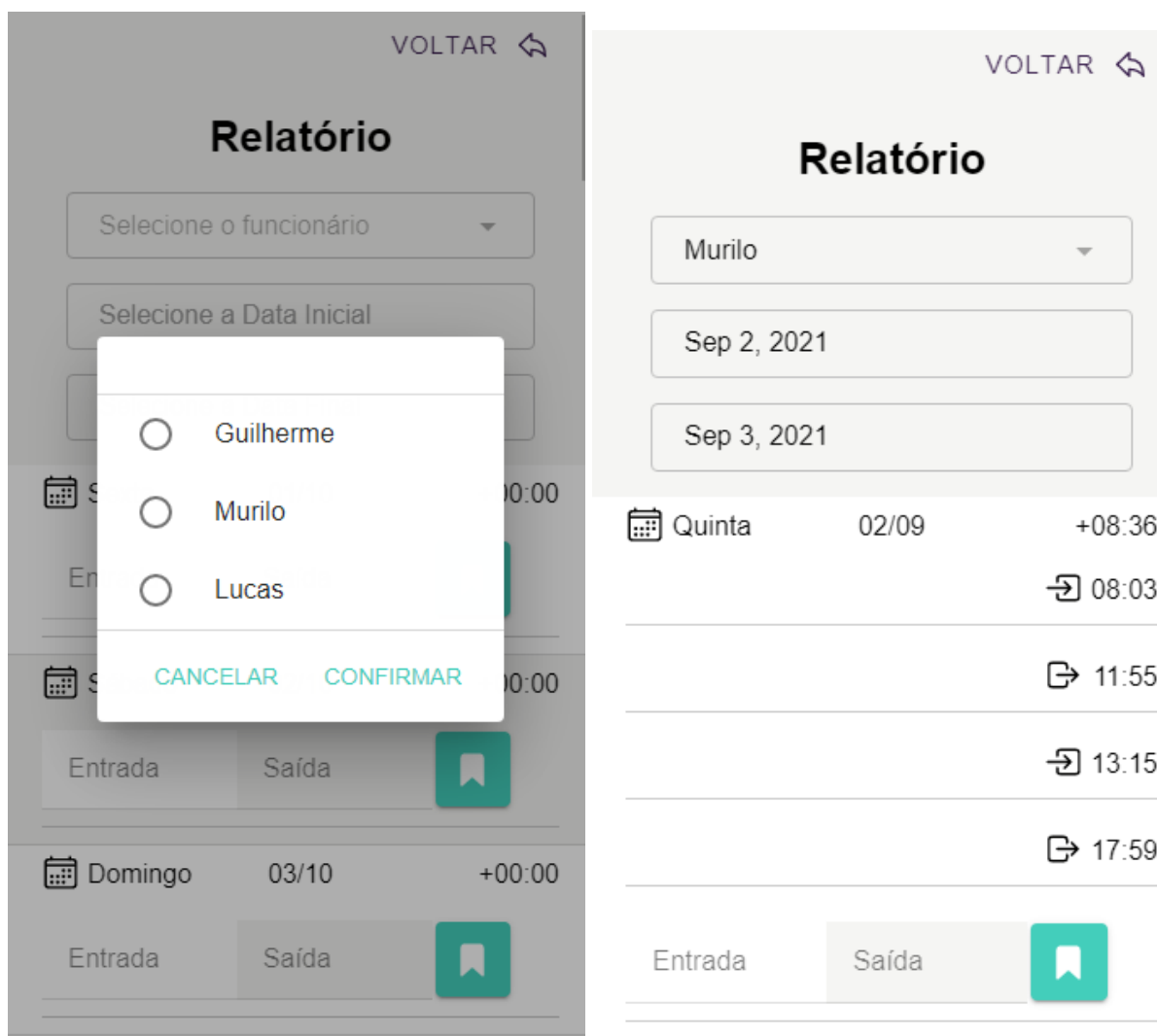
Horário de saída (Manhã)

Horário de Entrada (Tarde)

Fonte: (AUTOR, 2021)

A quinta e última opção, mostrada na Figura 12, é a opção "RELATÓRIO". Nesta tela o usuário administrativo pode selecionar qual funcionário ele deseja visualizar e em qual intervalo de tempo. Após a seleção, ele poderá visualizar todas as entradas e saídas com seus respectivos horários, mostrando o sinal positivo ou negativo referente ao seu saldo de horas. O cálculo do saldo é feito com base no horário atribuído ao funcionário. Registros feitos fora do horário designado são considerados para o saldo positivo, para que a empresa tenha conhecimento. As telas para o filtro e relatório, com exemplos de registros, podem ser vistas na Figura 17.

Figura 17 – Página de relatório de todos funcionários

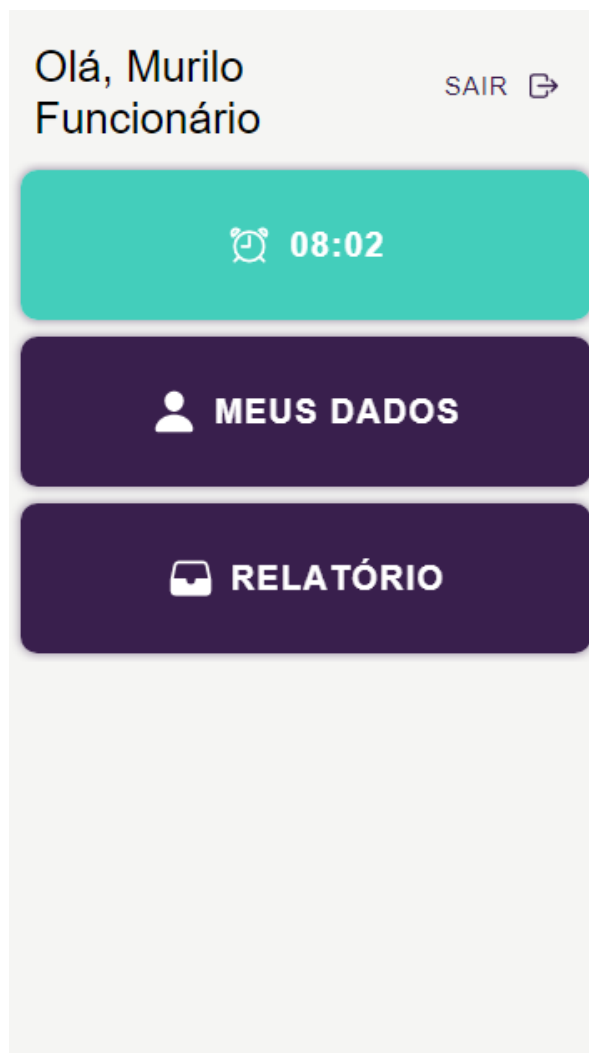


Fonte: (AUTOR, 2021)

4.2 Módulos do funcionário

O módulo de funcionamento para funcionários inicia-se com a página de *login*. Após a validação das credenciais e verificação do tipo de usuário entre administrador e funcionário, ele é direcionado para a tela principal. No módulo em questão, ele é direcionado para a tela principal de funcionário. Nessa tela o usuário tem acesso ao menu, o qual irá direcioná-lo para a atividade que deseja executar. Nessa tela também é possível ver, logo no início, o horário do último registro efetuado. Os detalhes podem ser vistos na Figura 18.

Figura 18 – Página de principal de funcionário



Fonte: (AUTOR, 2021)

Selecionando o segundo bloco, exibido na Figura 18, o bloco "MEUS DADOS", o funcionário é direcionado para a página onde pode editar as suas informações pessoais. Nessa tela é possível editar o seu nome, email, telefone para contato e *CPF*. A tela para a edição dos dados pode ser vista na Figura 19.

Figura 19 – Página de dados do funcionário



VOLTAR ↶

Meus Dados

Murilo Funcionário

func@geo.com

(54) 99999-9999

111.111.111-11

Salvar

Fonte: (AUTOR, 2021)

Selecionando o terceiro bloco visível na Figura 18, o bloco "RELATÓRIO", o funcionário é direcionado para visualizar todos os seus registros. Nessa tela, o funcionário não tem a permissão para editar um registro já efetuado ou lançar novos registros. Essa funcionalidade fica disponível apenas para usuários administrativos. O funcionário pode filtrar por datas o intervalo que deseja visualizar. O relatório mostra também qual é o saldo do funcionário em cada dia, com base no horário de trabalho que ele faz parte e um balanço geral das horas trabalhadas dentro do intervalo selecionado. Todas as informações podem ser melhor visualizadas na Figura 20.

Figura 20 – Página de relatório para o funcionário

Relatório		
Feb 10, 2021		
Feb 12, 2021		
📅 Quarta	10/02	-00:11
		↪ 08:10
		↪ 11:59
		↪ 13:15
		↪ 18:00
📅 Quinta	11/02	-00:19
		↪ 08:17
		↪ 11:59
		↪ 13:31

Fonte: (AUTOR, 2021)

Para a finalização das atividades o usuário, tanto administrativo como o funcionário, podem efetuar o *logout* do sistema. Clicando no botão superior direito na tela principal, o usuário encerra a sessão e é direcionado para a tela de *login*. O botão pode ser visto, para ambos os níveis de usuários, na Figura 12 e Figura 18.

5 CONCLUSÃO E TRABALHOS FUTUROS

O objetivo deste trabalho foi desenvolver um aplicativo designado ao gerenciamento de registros de horário ponto dentro de uma empresa. As primeiras etapas para realizar o processo se caracterizaram pelas análises de ferramentas existentes no mercado, como auxílio para efetuar os escopos e requisitos para o desenvolvimento do trabalho. Com a análise, foi possível ver que muitas agências ainda utilizam sistemas antigos de gerenciamento de registro. Estes sistemas ainda utilizam uma grande quantidade de impressão de papéis desnecessária, além de um custo elevado na instalação de aparelhos nas dependências da empresa, o deslocamento dos funcionários até o local de registro e a manutenção corriqueira que é gerada. Sendo assim, deu-se sequência ao desenvolvimento deste trabalho, para promover uma ferramenta que pudesse agilizar o processo, utilizando as tecnologias presentes hoje nos smartphone, evitando a impressão de papéis para validar o registro, reduzindo custos de manutenção e instalações, proporcionando um rápido acesso dos relatórios em tempo real e um controle prático dos funcionários que utilizarão a ferramenta.

O segmento que denotou maior esforço no desenvolvimento do trabalho foi referente à construção dos registros de ponto, utilizando a geolocalização. A partir dele, é possível gerar todos relatórios que são a peça chave para um bom resultado final do aplicativo. Por se tratar de uma etapa bastante sensível em relação ao funcionamento da ferramenta, foram desenhados alguns esboços para que fosse possível tomar a melhor decisão de qual método a ser empregado. Os estudos obtiveram um ótimo resultado.

Uma das características do vigente trabalho, foi o seu desenvolvimento utilizando o *framework Ionic*, que auxiliou para que o trabalho fosse desenvolvido com a utilização de poucos recursos. Juntamente com o levantamento dos requisitos, foi possível a construção do mesmo de forma incessante.

Os trabalhos futuros propostos para este aplicativo tem a finalidade de torná-lo o mais prático do mercado, e que atenda todos os níveis de usuários sem a comunicação por outros meios que não seja a ferramenta em questão. Para a praticidade, é necessário que seja estudado os sistemas operacionais de celulares, pois o sistema criado não é eficaz quando o aparelho não o está executando em primeiro ou segundo plano. Outra melhoria de bastante relevância é a situação onde funcionários querem alterar um registro de horário. Nesse caso, deverá ser feita uma solicitação de alteração e aprovada pelo usuário administrador da empresa.

O vigente trabalho conta com uma grande gama de melhorias futuras que o tornariam ainda mais robusto e completo. Junto à *API* desenvolvida, fica de fácil acesso a elaboração de uma aplicação *web* para atender usuários que queiram registrar o seu ponto pelos computadores pessoais e para usuários administrativos uma melhor visibilidade de seus relatórios, além de possíveis integrações com sistemas de terceiros.

Sob esse prisma, conclui-se que os requisitos estabelecidos no projeto de conclusão

de curso e na execução do presente trabalho foram aplicados em seu desenvolvimento, desta maneira, alcançando a finalidade proposta.

REFERÊNCIAS

- DEVFURIA. **O que é php?** 2020. Disponível em: <<http://devfuria.com.br/php/o-que-e-php/>>. Acesso em: 07 de agosto de 2021. Citado na página 14.
- GAME, A. **O que é JSON.** 2011. Disponível em: <<https://www.devmedia.com.br/o-que-e-json/23166>>. Acesso em: 13 de setembro de 2021. Citado na página 18.
- HIGA, P. **O que é XAMPP e para que serve.** 2012. Disponível em: <<https://www.techtudo.com.br/dicas-e-tutoriais/noticia/2012/02/o-que-e-xampp-e-para-que-serve.html>>. Acesso em: 23 de agosto de 2021. Citado na página 18.
- JUNIOR, R. **Advertência por deixar de marcar ponto, é legal?** 2020. Disponível em: <<https://www.jornalcontabil.com.br/advertencia-por-deixar-de-marcar-ponto-e-legal/>>. Acesso em: 28 de agosto de 2021. Citado na página 13.
- MEDEIROS, H. **Application Programming Interface: Desenvolvendo APIs de Software.** 2014. Disponível em: <<https://www.devmedia.com.br/application-programming-interface-desenvolvendo-apis-de-software/30548>>. Acesso em: 21 de agosto de 2021. Citado na página 18.
- PACHECO, R. A. F. **Heroku: o que é e como funciona.** 2020. Disponível em: <https://blog.geekhunter.com.br/heroku/#O_que_e_Heroku>. Acesso em: 23 de agosto de 2021. Citado na página 19.
- POSTGRESQL. **PostgreSQL: About.** 2021. Disponível em: <<https://www.postgresql.org/about/>>. Acesso em: 16 de agosto de 2021. Citado 2 vezes nas páginas 15 e 16.
- VASCONCELOS, E. **Registro de ponto: Saiba o que é, o que diz a lei e como reduzir custos com essa ferramenta.** 2021. Disponível em: <<https://www.jornalcontabil.com.br/registro-de-ponto-saiba-o-que-e-o-que-diz-a-lei-e-como-reduzir-custos-com-essa-ferramenta/>>. Acesso em: 14 de dezembro de 2021. Citado na página 22.
- ZANETTI, H. P. O. **Javascript Descomplicado Programação para web, IOT e dispositivos móveis.** [S.l.]: Saraiva, 2020. Citado na página 17.