

UNIVERSIDADE PAULISTA – UNIP  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

**SOLUÇÃO WEB PARA ADMINISTRAÇÃO E GESTÃO DE IGREJAS COM  
AUXÍLIO DE RECONHECIMENTO FACIAL**

LUCAS NONATTO

LIMEIRA-SP, NOVEMBRO DE 2019

LUCAS NONATTO

**SOLUÇÃO WEB PARA ADMINISTRAÇÃO E GESTÃO DE IGREJAS COM AUXÍLIO  
DE RECONHECIMENTO FACIAL**

Trabalho de Conclusão de Curso para obtenção do título  
de Bacharel em Ciência da Computação pela  
Universidade Paulista.

Professores Orientadores: M. Sc. Marcos Vinícius Gialdi  
e M. Sc. Mateus Locci

LIMEIRA-SP, NOVEMBRO DE 2019

LUCAS NONATTO

**SOLUÇÃO WEB PARA ADMINISTRAÇÃO E GESTÃO DE IGREJAS COM  
AUXÍLIO DE RECONHECIMENTO FACIAL**

---

Prof. Marcos Vinícius Galdi

---

Prof. Mateus Locci

Dedico este trabalho primeiramente a Deus,  
meu pai, aos meus colegas, todos os professores  
para a realização de mais uma etapa.

## RESUMO

Por meio deste TCC um ERP 100% na web utilizando principalmente a tecnologia Code Igniter, usando também banco de dados Mysql para que seja possível gravar informações como, cadastro de usuário, membros, entre outros além disso terá relatórios para que os usuários possam ver por exemplo os aniversariantes do mês, os lançamentos de presença da célula, o diferencial do sistema é que nele há uma ferramenta que será possível identificar os membros que estão chegando pelos portões principais, claro com o auxílio de câmeras posicionadas em pontos estratégicos, com o reconhecimento facial é possível fazer várias coisas como lista de presença e até reconhecimento facial tornando-se possível identificar os membros que estão comparecendo as reuniões e eventos da igreja, hoje em dia existem poucos sistemas que fazem isso, atualmente temos sistemas para igrejas, mas geralmente são para instituições de grande porte, a ideia com esse sistema é atingir todo tipo de público, seja pequena ou grande, assim é possível ter um controle da igreja muito maior independente da verba disponível.

Palavras chaves: Inovação; Facilidade; Confiança; Prático; Funcional.

## ***ABSTRACT***

*On this document, a web system made using the framework Code Igniter, also using Mysql database to be able to record information such as user registration, members, among others will also have reports to users consult, like the birthdays of the month, the events presence launches, the new concept on this system is that there is a tool in it that will be able to identify the members that are coming through the main gates, of course with the help of cameras positioned at strategic points, with the facial recognition it's possible do a lot of things like attendance list and making possible to identify the members who are attending church meetings and events, there are few systems that do that, we currently have church systems, but they are usually for large institutions, the idea with this system is to reach all kinds of church, small or large, it is possible to have much greater church control regardless of the budget available.*

*Keywords: Innovation; Facility; Confidence; Practical; Functional.*

## SUMÁRIO

|  |    |
|--|----|
| INTRODUÇÃO.....  | 10 |
| 1.1 Objetivo.....  | 10 |
| 1.2 Justificativa.....                                     | 11 |
| 1.3 Metodologia.....                                       | 11 |
| 1.4 Pressupostos da Pesquisa.....                          | 12 |
| 1.5 Método Científico Utilizado.....                       | 12 |
| 1.6 Estrutura do Trabalho.....                             | 13 |
| 2.0 O que é ERP em nuvem?.....                             | 13 |
| 2.1 Porque utilizar em nuvem?.....                         | 13 |
| 2.2 Modelos de negócio de <i>cloud computing</i> .....     | 14 |
| 2.3 Reconhecimento Facial.....                             | 14 |
| 2.4 Reconhecimento Facial pode invadir a privacidade?..... | 15 |
| 3.0 Problemas para reconhecimento facial.....              | 15 |
| 3.1. Reconhecimento facial com <i>python</i> .....         | 15 |
| 4.0 Utilizando CodeGinter.....                             | 16 |
| 4.1 Explicando na prática.....                             | 16 |
| 4.2 Padrão MVC.....  | 18 |
| 4.3 Por que usar MVC?.....                                 | 18 |
| 5.0 Desenvolvimento do sistema.....                        | 19 |
| 5.1.1. Cadastro de Membro.....                             | 19 |
| 5.1.2. Cadastro de Usuário.....                            | 22 |
| 5.1.3. Cadastro de Permissão.....                          | 22 |
| 5.1.4. Cadastro de Célula.....                             | 23 |
| 5.2.1. Lançamento de Presença.....                         | 24 |
| 5.2.2. Lançamento de Agenda.....                           | 25 |
| 5.3.1. Relatório de Aniversariantes.....                   | 26 |
| 5.3.2. Relatório de Presença.....                          | 26 |
| 5.3.3. Relatório de Membro.....                            | 26 |
| 5.4. Tela Inicial.....                                     | 27 |
| 5.5. Considerações Finais.....                             | 28 |
| 6.1. Uso de Python.....                                    | 28 |
| 6.2. Escaneando.....                                       | 28 |
| 6.3. Aprendizagem.....                                     | 29 |
| 6.4. Conexão e envio de dados.....                         | 30 |
| 6.5. Conclusão Python.....                                 | 31 |
| 7.0 Conclusão.....   | 31 |
| 8.0 Referências Bibliográficas.....                        | 32 |

| <b>ABREVIATURAS E SIGLAS</b> |  |
|------------------------------|--|
| BD                           | Banco de Dados   |
| ERP                          | <i>Enterprise Resourcing Planning</i> (Planejamento de Recursos para a Empresa)                    |
| JAVA                         | Linguagem de Programação Orientada a Objetos   |
| MVC                          | <i>(Model-View-Control)</i>  |
| SQL                          | <i>Structured Query Language</i> (Linguagem de consulta estruturada)                               |
| PHP                          | Um acrónimo recursivo para " <i>PHP: Hypertext Preprocessor</i> " - é uma linguagem de programação |
| SEBRAE                       | Serviço Brasileiro de Apoio às micro e pequenas empresas   |
| SENAI                        | Serviço Nacional de Aprendizagem Industrial  |
| TI                           | Tecnologia da Informação   |
| SaaS                         | <i>Software as a Service</i>   |
| IaaS                         | Infrastructure As a Service  |
| CI                           | CodeIgniter  |
| HTML                         | <i>Hypertext Markup Language</i> - Linguagem de Marcação de Hipertexto                             |



## FIGURAS

|  |    |
|--|----|
| Figura 1 – Tela de lista de membros do sistema eHoly.....                      | 16 |
| Figura 2 – <i>Select</i> para carregar resultados na tela de Membros.....      | 17 |
| Figura 3 – <i>Controller</i> para chamar o <i>select</i> do <i>Model</i> ..... | 17 |
| Figura 4 – Tela do sistema eHoly (Lista de Membros).....                       | 20 |
| Figura 5 – Tela do sistema eHoly (Evento para Membro).....                     | 20 |
| Figura 6 – Tela do sistema eHoly (Cadastro).....                               | 21 |
| Figura 7 – Tela do sistema eHoly (Histórico).....                              | 22 |
| Figura 8 – Tela do sistema eHoly (Cadastro Usuário).....                       | 22 |
| Figura 9 – Tela do sistema eHoly (Permissões).....                             | 23 |
| Figura 10 – Tela do sistema eHoly (Cadastro Célula).....                       | 24 |
| Figura 11 – Tela do sistema eHoly (Lançamento Célula).....                     | 25 |
| Figura 12 – Tela do sistema eHoly (Lançamento Agenda).....                     | 25 |
| Figura 13 – Tela do sistema eHoly (Filtro relatório membro).....               | 26 |
| Figura 14 – Tela do sistema eHoly (Tela Inicial).....                          | 27 |
| Figura 15 – Código Python (Script para escaneamento).....                      | 29 |
| Figura 16 – Código Python (Script para aprendizado).....                       | 30 |
| Figura 17 – Código Python (Script para inserção).....                          | 31 |

## INTRODUÇÃO

Os estudos acerca de criar um ERP 100% Web para que seja de fácil acesso para que o possuir, que possa, por exemplo, estar em sua casa e consiga ver o relatório tenha caso esquecido de ver quando estava no local com computador, ou seja, ganhamos facilidade e pró atividade tendo o sistema na web, o que é o mais importante para um sistema é a funcionalidade e a facilidade de fazer as operações e consultas de onde estiver, nos dias de hoje o que mais queremos é ter tudo o mais fácil e disponível possível em nossas mãos e a ideia desse sistema é realmente essa, ter todas as informações em nossas mãos.

Os pastores das igrejas que utilizarem esse sistema poderão caso queiram cadastrar membros, mais usuários, se por acaso tenham esquecido de fazer isso durante estava no local com computadores da igreja.

Um dos maiores desafios ao gerenciar uma igreja é manter os membros delas sempre próximos, com o sistema ficará mais fácil uma vez que é possível ter um relatório de presença dos cultos, células ou até dos aniversariantes do mês.

Para ter todos os membros da igreja sempre por perto é importante que o pastor de o máximo de atenção para todos, só que muitas vezes é difícil saber todos os dados, quem está ou não vindo as reuniões, então para que ajudar com essas questões que o sistema vem, com o cadastro com vários dados importantes o líder sempre saberá as informações de cada membro.

Com o reconhecimento facial que tudo muda de figura, não é possível ter o controle de todas as pessoas que estão comparecendo as reuniões, dependendo da quantidade de gente que frequenta a igreja diariamente, mas com o lançamento de presença automático graças a ajuda do reconhecimento facial utilizado com *python*, será possível ter um relatório para saber quem veio ou não, e se o líder desejar poderá entrar em contato com o indivíduo que não vai até a igreja a algum tempo.

### 1.1 Objetivo

O maior desafio é sanar um problema que existe em praticamente em todas as igrejas do mundo inteiro, que é a proximidade dos membros com os líderes, como por exemplo o membro, muitas vezes precisam de uma visita, pois está passando por um momento difícil e acaba deixando de frequentar as reuniões, mas por ter um número muito elevado de membros o líder da igreja pode acabar não notando que o membro deixou de ir

e acaba ficando por isso mesmo, o membro para de ir na igreja e continua com o problema, com o sistema vai ser possível que o líder note que o membro está deixando de ir nas reuniões e com os cadastros de membro atualizados será informado o número celular, endereço, onde ele poderá mandar alguém ou até mesmo ele ir até a residência do membro, atualmente os líderes tentam fazer todo o controle apenas anotando ou até em casos mais raros tentando armazenar todas as informações apenas na memória, o que é muito difícil. A maneira que o sistema fará isso é com câmeras posicionadas em pontos da igreja identificando quem está chegando, no final do mês o líder poderá consultar um relatório dos membros que não compareceram aos cultos, deixar isso de forma fluida e orgânica é essencial, pois o usuário tem que se sentir confortável usando o sistema.

## **1.2 Justificativa**

É importante que quando o projeto ficar pronto ajudará muito os líderes de igreja, ajudando-os mostrando como pode ser mais fácil e simples ter mais controle do seu trabalho e talvez quem sabe ganhar mais membros por causa do sistema, uma vez que cuidando bem dos

membros que ele tem, eles poderão chamar mais gente para frequentar a igreja, eles poderão ver informações de presença ou até de aniversariantes do mês nos relatórios que estarão disponíveis no sistema, com essas informações com certeza o líder tomará decisões importantes para melhor administrar sua igreja, atualmente muitos sistemas o usuário tem apenas informações fixas sobre o membro, mas com a ideia desse TCC será possível gerenciar e administrar a igreja de forma muito mais eficaz e prática.

## **1.3 Metodologia**

A primeira etapa consiste em conversar com algum líder de igreja para saber os pontos mais importantes para o cadastro de um membro, imaginando que as principais informações são, nome, telefone e endereço que serão os meios de se comunicar com o membro. Depois disso, o banco de dados terá que ser desenvolvido, usando MYSQL, pois é a ferramenta que melhor se adapta para o modelo que será apresentado futuramente, 100% web. Com o banco de dados feito, será necessário fazer todo sistema, utilizando a framework Code Igniter, que usa a linguagem PHP para codificação, depois de todo o site feito, com Bootstrap, CSS, JavaScript e JQuery para que de todo um *layout* atual para o sistema, será feitos testes no sistema todo, depois será entregue para que um líder de igreja

possa usá-lo, ele utilizando-o sendo possível sugestões para melhorar cada vez mais o sistema, com tudo finalizado, parte de *layout*, programação, banco de dados, testes, chega a hora de partir para a cereja do bolo, que é o reconhecimento facial, será usado a ferramenta python em um minicomputador, onde ele rodará os arquivos que serão executados de forma automática quando chegar uma hora do dia ou semana, desenvolvido toda a parte de reconhecimento facial, agora é preciso focar na conexão com o banco de dados e a inserção dos dados importantes retirados pelo reconhecimento facial, feito os testes e funcionando tudo seguimos para a implementação, na hora da implementação será preciso escanear os rostos de todos os membros da igreja, então ensinar a inteligência artificial que aqueles rostos são válidos para inserir no banco de dados e por fim, fazer um teste de campo onde será feita uma reunião na igreja e o sistema informará quais foram os membros que foram até lá naquele dia, lembrando que essa parte de implementação é necessário um investimento, pois será possível apenas fazer os testes em ambiente controlado até lá.

#### **1.4. Pressupostos da Pesquisa**

Apesar dos amplos recursos avançados de computação, podem existir líderes que ainda colocam barreiras para utilização de tecnologias. Portanto, alguns pressupostos podem ser levantados para investigação e eventual comprovação:

- ⌚ O líder da igreja acredita não precisar da informática, argumentando que consegue controlar e administrar a empresa por meio de processos manuais
- ⌚ O líder da igreja desconhece tecnologia e informática em geral

#### **1.5. Método Científico Utilizado**

O estudo em base dessas tecnologias busca sanar toda qualquer dificuldade que os líderes religiosos possam ter ao tentar administrar sua igreja. Com base nos estudos será possível criar uma ferramenta útil para ajudar-lhos da melhor forma possível.

Foi feito um estudo de campo para que o sistema possa atender da melhor forma possível qualquer tipo de igreja, seja ela qual denominação for, a tecnologia pode auxiliar a todos, mas o estudo foi feito com base nas necessidades principais de um líder religioso evangélico, sendo que a maior dificuldade dele foi ter os dados de todos os membros e que seja fácil consultá-los no dia a dia, além da necessidade de relatórios que enriquecem ainda mais o sistema, pois com eles fica ainda mais nítido os resultados, de acordo com o líder religioso.

## **1.6. Estrutura do Trabalho**

A introdução apresenta os tipos de Softwares Livres aplicados nos processos administrativos, a contextualização do tema, propõe o objetivo geral e o específico, elaborando, o método científico utilizado e a estrutura do trabalho.

Capítulo 2: descreve realmente ERP em Nuvem e reconhecimento facial.

Capítulo 3: problemas para o reconhecimento facial e suas dificuldades

Capítulo 4: CodeIgniter e sua aplicação

Capítulo 5: desenvolvimento do sistema e seus módulos

Capítulo 6: aplicação do Python

Capítulo 7: Conclusão

A conclusão fecha o trabalho revisando os resultados de campo com o sistema parcialmente finalizado e os resultados da ferramenta de reconhecimento facial e seus resultados.

## **2. O que é ERP em nuvem?**

Basicamente um ERP em nuvem é um sistema 100% na internet, o que pode trazer vários benefícios que serão comentados posteriormente nesse documento. Um ERP em nuvem é utilizado geralmente em casos que o cliente deseja usá-lo de qualquer lugar do mundo, pois com o sistema em nuvem é possível acessá-lo de apenas tendo conexão com internet e um dispositivo que tenha acesso a um navegador, tendo essas duas coisas já é possível usar o sistema, o que dá uma agilidade muito grande para clientes que precisam consultar relatórios importantes de uma hora para outra.

No caso do sistema para a igreja será necessário ser em nuvem, pois muitas vezes o líder precisará consultar pelo celular o relatório de presença ou o de aniversariantes do mês, com o sistema sempre disponível em qualquer lugar ele poderá ter essas informações a qualquer hora, sendo assim uma ferramenta muito importante para gerenciar a igreja.

### **2.1. Porque utilizar um sistema em nuvem?**

O sistema em nuvem deve ser utilizado por que todas as grandes empresas estão migrando para sistema web, a SAP mesmo esta migrando todo o seu software para nuvem, pois eles estão vendo que seus clientes não querem mais ficar dependendo de instalações,

ou ficar preso a um único computador sendo que pela web é possível acessar o sistema até pelo smartfone.

Whatsapp, Facebook, Instagram, Google Drive e Drop Box são exemplos de empresas que oferecem software como serviço ou SaaS(*Software as a Service*) e que são exemplo de sucesso no meio empresarial, pois eles revolucionaram e inovaram utilizando o sistema diretamente na web e hoje são marcas poderosas, pois tomaram a iniciativa e ousaram apostando nessa ferramenta incrível que é a web.

## **2.2. Modelos de negócio de *cloud computing***

- Software As a Service(SaaS) é o tipo de negócio que vende o software como serviço, como seu próprio nome diz, ou seja, o sistema é vendido e tem uma taxa mensal para ser mantida em posse do cliente, muitas vezes é feito um acordo para que essa taxa mensal acompanhe um suporte com um limite de horas no mês. Geralmente esse modelo de negócio é visto com bons olhos, pois o desenvolvedor tem uma renda mensal e para o cliente também por que eles pagam apenas pela utilização, que normalmente é medida por quantidade de usuários.
- Infrastructure As a Service(IaaS) é o modelo de negócio utilizado para venda de recursos computacionais, como por exemplo, mais espaço de armazenamento e capacidade de rede, sob uma taxa mensal da utilização do serviço.
- Plataforma As a Service (PaaS) é quando o fornecedor vende uma plataforma de desenvolvimento de software, que pode ser usada em vários lugares diferentes, ou apenas em um, para desenvolver o PaaS deve-se usar bibliotecas de uma determinada plataforma. O que acaba se tornando possível a escalabilidade do software.

## **2.3. Reconhecimento Facial**

Reconhecimento facial é a tecnologia que detecta o rosto de uma pessoa e compara com seu banco de dados para assim então informar quais as informações daquela pessoa que foi detectada pela câmera.

O reconhecimento facial atualmente está sendo usado de maneira mais efetiva pelo Facebook, eles fazem de uma maneira que sem perceber tem uma tecnologia fascinante bem na palma de nossas mãos, eles usam reconhecimento facial assim quando você entrar na foto as pessoas que estão naquela foto, caso elas estejam no seu grupo de amigos da plataforma, será marcado automaticamente a pessoa da foto. E essa é apenas umas das possíveis utilidades do reconhecimento facial, no metro de São Paulo foram instaladas portas interativas que exibem anúncios e detectam as reações das pessoas que estão passando e olham para o letreiro, a tecnologia é capaz de detectar o sexo e a idade da pessoa, além disso, é possível saber se a pessoa ficou feliz, insatisfeito, surpreso ou neutro em relação a propaganda.

## 2.4. Reconhecimento Facial pode invadir a privacidade?

Até certo ponto algumas pessoas podem ficar incomodadas com uma câmera filmando-as e pegando informações, mas para o caso desse projeto isso não será um problema, pois os membros da igreja entenderão que essa tecnologia será apenas para controlar a presença deles no mês, com uma conversa com um líder de igreja foi questionado isso e ele respondeu exatamente isso, e que caso ele tenha algum problema de algum membro com relação a isso ele explicará novamente e em caso mais extremos poderá tirar a pessoa do cadastro do sistema, para que o sistema não detecte mais a pessoa que está desconfortável com o reconhecimento facial.

## 3. Problemas para reconhecimento facial?

Um dos problemas que gera mais discussão quando se trata de reconhecimento facial em lugares que tem uma transição grande de pessoas é a questão de direitos de liberdade dos civis, uma vez que terá câmeras posicionadas monitorando e escaneando os rostos de qualquer pessoa que passar por ali.

Recentemente saiu uma notícia via site do G1, que o São Francisco proibirá o uso de reconhecimento facial pela polícia, por justamente afetar o direito de liberdade dos cidadãos da cidade, na matéria estava reportando que oito dos nove membros do conselho nacional votaram a favor da proibição das câmeras usadas pela polícia, uma informação importante é que essa proibição não valerá para aeroportos e instalações do governo federal dos Estados Unidos, ou seja, por enquanto somente a polícia não poderá utilizar essa ferramenta na ajuda contra o crime. Mas só porque São Francisco está proibindo o reconhecimento facial, não quer dizer que outros governos aderirão a essa proibição, uma vez que o reconhecimento facial pode ajudar não só polícias mas também em hospitais, por exemplo, uma câmera em cada quarto de paciente que detecta qualquer alteração do paciente e emite um sinal no computador das enfermeiras, isso poderá salvar vidas, ou seja, essa proibição provavelmente não será uma grande interferência, pois o reconhecimento facial traz mais benefícios do que mais malefícios.

De acordo com a matéria também algumas organizações como a *Stop Crime SF* se manifestaram dizendo que o reconhecimento facial vem mais para ajudar do que atrapalhar, que a tecnologia pode auxiliar na busca de crianças perdidas, pessoas com demência e até o tráfico sexual, outra organização que se manifestou foi a União Americana da Liberdade Civil disse que ao contrário do que o governo de São Francisco acha a tecnologia pode ser usada de maneira passiva e não exigir o conhecimento, consentimento ou participação do indivíduo. Uma boa notícia para a tecnologia de reconhecimento facial.

### 3.1. Reconhecimento facial com *python*

Para esse protótipo foi pensado que a melhor alternativa seria usar a ferramenta *python*, além dela ser uma tecnologia que permite fazer várias interações com imagens e vídeos é uma linguagem relativamente simples para o desenvolvedor se comparado com outras. Está sendo utilizado a biblioteca com código aberto *OpenCV*, dentro dela possui vários tipos de auxiliares para vários tipos de projeto, seja ele para processamento de imagem, análise de vídeo, detecção de objetos, entre outras.

O código desenvolvido é relativamente simples, pois todo o peso para que tudo dê certo fica com a biblioteca *OpenCV*, para que o reconhecimento de certo, é preciso fazer

um pré escaneamento de todos os rostos que passarão pelas câmeras e que o líder queira que apareça na lista de presença, o código pergunta qual o código deseja ser vinculado aquele rosto e depois ira tirar cinquenta fotos do rosto da pessoa e armazenará dentro de uma pasta no computador, com o código informado no nome de cada foto, será necessário ensinar a inteligência artificial que aqueles são rostos válidos, então para isso, é executado outro código que ensina todos os rostos e armazena algumas informações em arquivos xml, depois de tudo isso, é feita a conexão com o banco de dados do sistema web e então será monitorado pela câmera qual são os rostos que estão passando pelo local, é feito um filtro dentro do código para que se aquela pessoa já foi detectada anteriormente, caso sim, não é inserido no banco de dados, se não foi visto ainda, é inserido e um detalhe importante é que só é inserido se a confiança da inteligência artificial for maior que quarenta por cento.

#### 4. Utilizando CodeIgniter

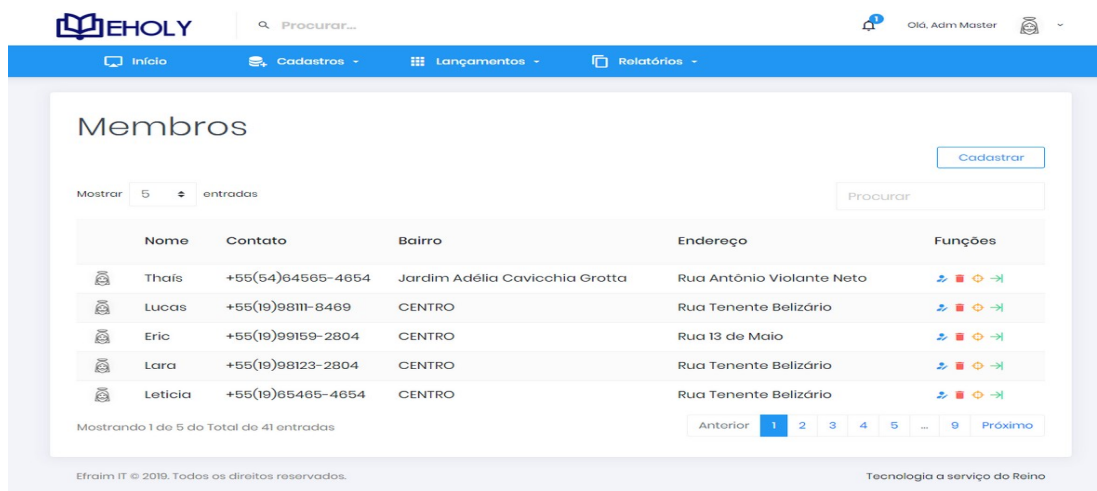
*CodeIgniter* que pode ser abreviado por CI, é uma *framework* PHP que usa o modelo MVC (*Model, View, Controller*), que basicamente, o *Model* é feito toda a parte de interações com o banco de dados, a *View* é a parte em que aparece para o usuário, ou seja, todo *HTML* é feito nessa parte e finalmente o *Controller* que é o “coração” do código, onde é juntado o *Model* e *View* e também pode ser programado algumas coisas em PHP para situações específicas.

Toda documentação do CI está no site oficial deles, nessa documentação explica tudo que é necessário para utilizar a framework, desde de como baixar e instalar o modelo MVC até como fazer as interações com o banco de dados no *Model* com as formas do CI.

##### 4.1. Explicando na prática

Para fazer uma tela será necessário pelo menos três arquivos, como qualquer framework que trabalha com o modelo MVC, primeiramente pode ser desenvolvido todo o *HTML* do projeto e esse código ficará na *View*, nessa parte do código pode ser utilizado qualquer coisa para auxiliar na parte *layout*, para o sistema da igreja será utilizado: *HTML, Bootstrap, JQuery, JavaScript, CSS e PHP*. Para ter uma ideia de como pode ficar a tela utilizando todas essas ferramentas temos uma imagem demonstrando:

Figura 1 – Tela de lista de membros do sistema eHoly



Fonte: Arquivo Pessoal



Para que sejam mostrados os resultados do banco de dados é necessário primeiro montar no *Model* o *select* que pode tanto ser feito na linguagem normal do banco de dados, ou então no modelo CI, para este exemplo preferi utilizar a linguagem normal do banco de dados:

Figura 2 – *Select* para carregar resultados na tela de Membros

```
public function membros() {
    $sql = "
    SELECT
    codigo,foto,nome,sobrenome,endereco,bairro,numero,complemento,numero_telefone,numero_celular,numero_trabalho
    FROM
    membro
    WHERE
    membro.data_exclusao IS NULL;
    ";

    return $this->db->query($sql)->result();
}
```

Fonte: Arquivo Pessoal

Agora que temos o resultado do *select* carregado no sistema, é necessário carregar ele na tela, para fazer isso, no *Controller* é preciso chamar a função que foi feita no *Model* com o *select* para tal necessidade. No *Controller* cada função pode ser feita uma nova página usando a mesma tela, por exemplo, para tela de cadastro e edição no sistema é usada exatamente o mesmo arquivo na *View* o que muda é a função chamada para mostrar os resultados ou então o campo em branco para que seja cadastrado um novo membro, mas neste exemplo será mostrado como é chamado a função do *Model* para mostrar os resultados carregados pelo *select*:

Figura 3 – *Controller* para chamar o *select* do *Model*

```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Membros extends CI_Controller {
5
6     function __construct(){
7         parent::__construct();
8         $this->load->model("Membros_model");
9     }
10
11     public function index()
12     {
13         $data['head'] = $this->load->view('layout/head', '', TRUE);
14         $data['footer'] = $this->load->view('layout/footer', '', TRUE);
15         $data['membros'] = $this->Membros_model->membros();
16         $this->load->view('membros/index', $data);
17     }
18 }
```

Fonte: Arquivo Pessoal

Reparando que primeiramente no construtor é necessário carregar todo o *Model*(Linha 8, Figura 3) para ai então ser carregado todas as informações da tela(Figura 3, Linha 13 e 14 são o layout de cabeçalho e rodapé do sistema, Linha 15 é o carregamento da função que foi criada no *Model* e Linha 16 é carregada a *View*, que foi codificada, com todas as informações de rodapé, cabeçalho e dados).

Basicamente para se programar usando CI é assim, obviamente que aqui foi explicado uma funcionalidade bem simples do sistema para que seja mais didático, mas com o decorrer do sistema é necessário fazer funções e lógicas mais complexas, como todo e qualquer projeto.

## 4.2. Padrão MVC

Utilizando o padrão MVC é utilizado em muitos projetos por causa de sua arquitetura, o que torna possível que seja que tarefas sejam divididas em camadas muito bem definidas, que são elas, já citadas anteriormente, *Model*, *View* e *Controller*, executa o que é definido e nada mais.

O padrão MVC trás como benefício isolar regras de negócio da apresentação do usuário final, sendo assim é possível utilizar as mesmas telas várias vezes alterando apenas a lógica do *Controller*. Uma das qualidades do padrão MVC é que ele pode ser usado em vários tipos de projeto, como desktop, mobile e web.

A comunicação entre interfaces e regras de negócios é definida através do *Controller* e ele que faz ser possível com que as camadas sejam separadas, quando uma ação é feita na interface, ela se comunica com o *Controller* e assim se comunica com o que for necessário para aplicação.

O *Model* por sua vez fica responsável com a parte de manipulação de banco ou até mesmo se o programador preferir poder fazer funções de calculo, dependendo do que o sistema precisar.

Tendo todas essas partes trabalhando juntas só ira faltar mostrar tudo para o usuário através da interface a *View*, onde ela se comunica com o *Controller*, que por sua vez pega as funções necessárias no *Model*, assim basicamente funciona o padrão MVC.

## 4.3. Por que usar MVC?

Além de agregar valor ao produto, visto que atualmente usar um padrão na hora de codificar é visto com bons olhos por clientes e grandes empresas, facilita e muito em futuras manutenções do sistema, uma vez que não é mais necessário ficar procurando em vários arquivos onde está algum problema, isso acontece por causa da separação das camadas, por exemplo, o sistema está dando um problema na visualização, o problema estará na *View*, se a *query* feita esta dando problema, será necessário ver o *Model*, enfim, com as camadas separadas fica mais fácil de saber onde procurar o erro.

Em pesquisas na internet existem algumas postagens que dizem que é complicado de dar manutenção em um sistema feito em MVC, mas uma vez que entender o padrão MVC e entender o código que estiver dando manutenção, facilita muito mais na hora de arrumar erro e falhas de lógica do que em uma aplicação sem estruturação nenhuma.

## 5. Desenvolvimento do Sistema

O desenvolvimento do sistema é utilizado, como dito anteriormente, a framework PHP CodeIgniter, que trabalha no padrão MVC, com isso facilita a manutenção e melhorias futuras do sistema.

Neste capítulo será apresentado as principais funcionalidades do sistema, o sistema é voltado na gestão de igrejas, auxiliar os líderes a ter os dados e informações pertinentes de todos os membros de sua denominação, com ferramentas intuitivas ficará mais fácil fazer a gestão da igreja. O sistema conta com alguns módulos de suma importância na hora de ter as informações dos membros e de gestão, são eles:

### **Cadastro**

- Membro;
- Usuário;
- Célula (será explicado quando for aprofundado esta parte do módulo);
- Permissão;

### **Lançamentos**

- Presença(será explicado quando for aprofundado esta parte do módulo);
- Agenda;

### **Relatório**

- Aniversariantes;
- Célula;
- Membro;

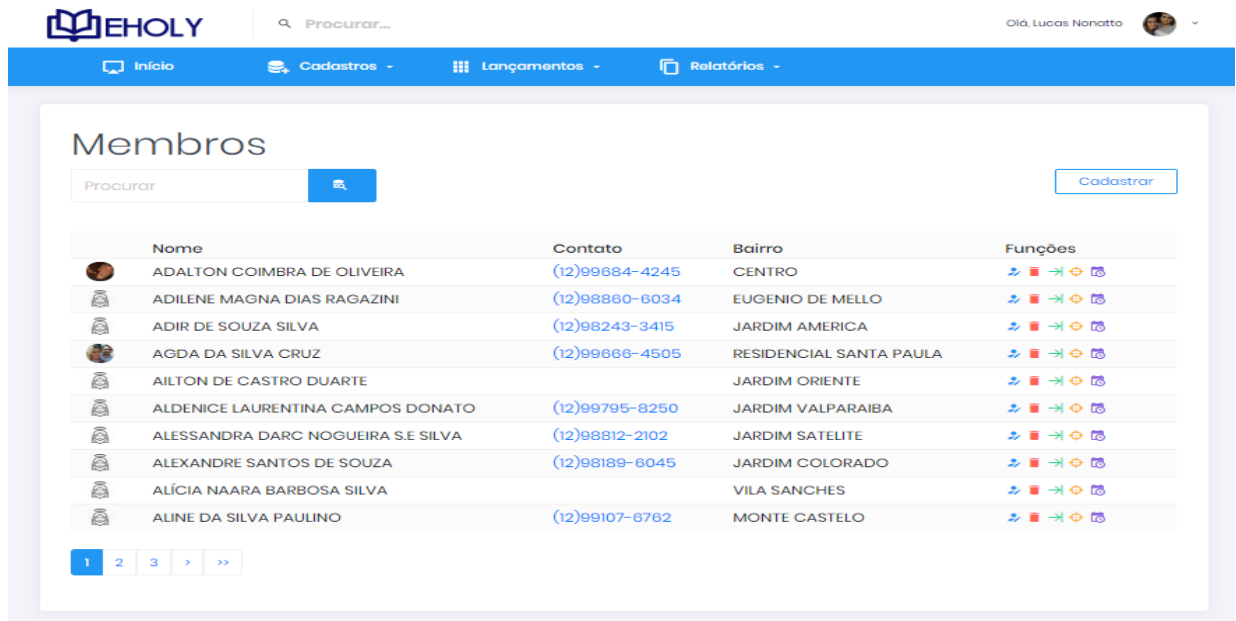
Até o momento são esses os módulos principais do sistema, a seguir será detalhado cada um deles.

#### **5.1.1. Cadastro Membro**

O cadastro de membro possui muitas informações, assim na hora que o usuário precisar saber alguma coisa sobre aquele membro terá muitos dados sobre ele, é mais trabalhoso na hora de cadastrar todos os membros da igreja, mas é um trabalho que irá valer a pena futuramente, uma vez que cadastrado todos os dados, além de ficar mais fácil para atualizações de informações, tudo que precisar saber sobre ele estará tudo em uma página web, que poderá ser acessada de qualquer lugar do mundo apenas com conexão de internet.

A seguir será mostrado as telas de lista e de cadastro dos membros, todos os dados que mostrados são fictícios e não revelam a verdadeira identidade dos nomes da lista ou do cadastro.

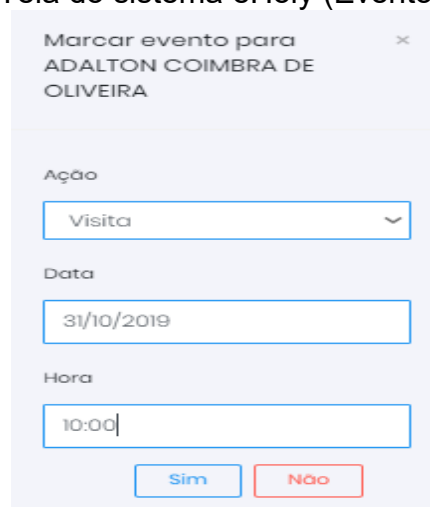
Figura 4 – Tela do sistema eHoly (Lista de Membros)



Fonte: Arquivo Pessoal

É possível ver que a tela é bem completa, com informações que são importantes que seja visto rapidamente pelo usuário, que são elas, nome, contato de celular e bairro. Mais a direita temos as funções para cada membro, o ícone azul edita, o vermelho apaga o membro da linha, o verde é uma carta de transferência, que serve para quando o membro vai se transferir de igreja, o líder precisa prover uma carta para outra instituição que o indivíduo irá se filiar, cada carta é única e contém informações dele, o botão amarelo funciona melhor quando o usuário está pelo celular, pois ao clicar no ícone o aplicativo do *Google Maps* é aberto automaticamente, com o endereço do membro como ponto de chegada e o ponto de partida é a localização atual do usuário, pelo computador funciona também, mas essa função é melhor utilizada com o celular, como dito anteriormente, para finalizar os ícones temos o roxo, que cria um evento para o membro da linha, uma imagem de exemplo:

Figura 5 – Tela do sistema eHoly (Evento para Membro)



Fonte: Arquivo Pessoal

Com essa funcionalidade é possível marcar um evento na agenda do usuário e será mostrado na tela inicial mostrando na agenda que aquele membro precisa de algum serviço do usuário em tal dia e em tal horário.

Uma das telas mais importantes do sistema inteiro é o de dados do membro, será mostrado a seguir os dados de um membro fictício:

Figura 6 – Tela do sistema eHoly (Cadastro)

Membros

Registro Histórico

Salvar Voltar

### Informações Pessoais

|              |                                   |                                   |
|--------------|-----------------------------------|-----------------------------------|
|              | Nome*<br>ADALTON                  | Data de Nascimento*<br>27/12/1967 |
|              | Sobrenome*<br>COIMBRA DE OLIVEIRA |                                   |
| Estado Civil | Solteiro(a)                       | Data do Casamento<br>dd/mm/aaaa   |
| Nome Pai     |                                   | Nome Mãe                          |
| Escolaridade | Nível Superior                    | Profissão<br>ENGENHEIRO ELÉTRICO  |
| RG           | 87.381.273-9                      | CPF<br>912.830.982-13             |




Fonte: Arquivo Pessoal

Nesta tela há muitos mais campos de informações de membro, mas para exemplo do sistema apenas esses serão o suficiente, nesta tela podemos ver que é disponível colocar uma foto do indivíduo sendo que tem a possibilidade de escolher uma foto do computador ou então tirar uma foto diretamente da *webcam* vinculada ao computador, essa foto ficará salva com nome do código do membro no servidor do sistema, assim não terá risco de ter duplicidade com o nome das imagens. Outro recurso da tela é imprimir todos os dados do membro em uma folha, clicando no ícone da impressora é aberta uma outra tela menor que é possível ver como ficará a impressão e escolher a impressora para ser usada.

Na aba histórico, é mostrado todos os eventos criados para esse membro, por exemplo se o membro recebeu uma visita, ou foi gerado uma carta de transferência, neste caso é disponibilizado um anexo do arquivo.

Figura 7 – Tela do sistema eHoly (Histórico)

Registro **Histórico**

| Data       | Ação          | Usuário       | Anexo   | Funções   |
|------------|---------------|---------------|---|---|
| 03/10/2019 | Transferência | Lucas Nonatto |  |  |
| 15/10/2019 | Visita        | Lucas Nonatto |   |  |

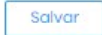

Fonte: Arquivo Pessoal


### 5.1.2. Cadastro Usuário

O cadastro de usuário é onde é definido o todas as informações do usuário do sistema e também a permissão de acesso que ele terá, a permissão de acesso será mais detalhada no próximo capítulo, é definido principalmente o nome de usuário e senha, que será usado para logar no sistema, segue a imagem da tela de cadastro:

Figura 8 – Tela do sistema eHoly (Cadastro Usuário)

Usuários



 



Nome\*  Email\*

Sobrenome\*  Usuário\*

Senha\*  Confirmar Senha\*  Perfil\*

Fonte: Arquivo Pessoal

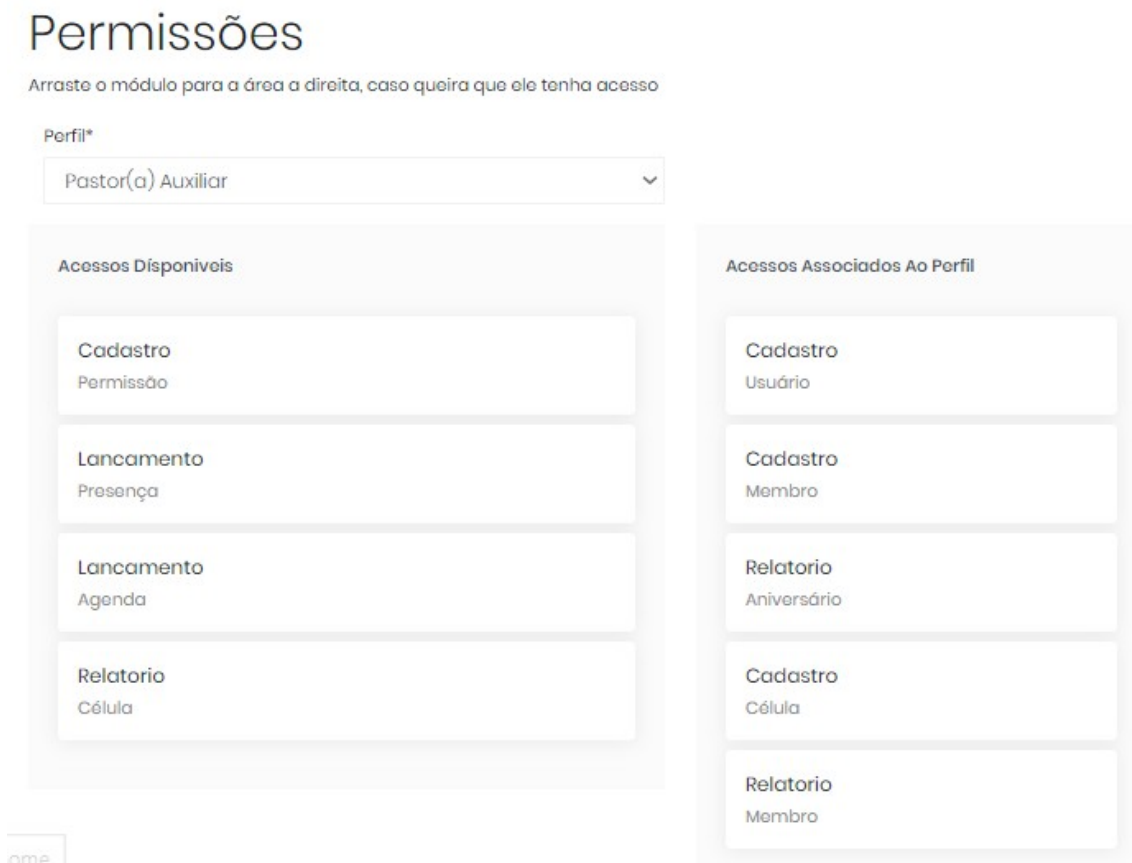
Assim como o cadastro de membro, a de usuário também possui a tela com a lista de usuários, mas elas são idênticas, com a única mudança que a de usuário tem apenas os ícones de apagar e editar. Podemos ver que uma vez definida o campo de usuário não é mais possível editá-lo, pois assim é possível manter o controle das ações do usuário dentro do sistema. O campo de perfil é pré setado no sistema, antes de fazer o cadastro dos usuários é necessário definir as permissões que esse perfil terá dentro do sistema.

### 5.1.3. Cadastro Permissão

O cadastro de permissão serve para que o sistema saiba quais módulos o perfil de acesso do usuário tenha disponível na hora que ele entrar no sistema. Por exemplo o pastor líder não quer que os outros usuários tenham acesso aos relatórios do sistema, então apenas arrastando os outros módulos para a direita ele dá permissão para que o perfil acesse apenas

os itens que estão na lista da direita e os da esquerda de maneira nenhuma o usuário que esta definido com aquele perfil de acesso possa acessar aquela parte do sistema. Imagem da tela de permissões:

Figura 9 – Tela do sistema eHoly (Permissões)



Fonte: Arquivo Pessoal

Tendo isso em mente o usuário principal não terá problemas na hora de definir quem poderá acessar o que no sistema, basta deixar os módulos que ele não quer que os outros usuários tenham acesso na lista da esquerda da imagem, assim os mesmos que tiverem aquele perfil de acesso não poderão acessá-los.

#### 5.1.4. Cadastro Célula

O cadastro de célula é uma das partes mais inovadoras do sistema inteiro, poucos sistemas têm uma funcionalidade igual a essa, pois é uma metodologia de célula é relativamente nova nas igrejas, então poucos gestores possuem essa função de cadastramento.

Com dados de endereço e pessoas importantes dentro da célula formam um cadastro bem completo do que hoje em dia é necessário para as igrejas crescerem cada vez mais essa metodologia. Uma ferramenta que é utilizada nessa tela é o *auto complete*, que no caso do endereço basta digitar o CEP da célula que ira trazer as informações complementares, para fazer isso funcionar é utilizado as ferramentas JQuery e AJAX, para que não seja necessário recarregar a página para inserir essas informações dentro dos campos. É utilizado essas ferramentas também nos campos de pessoas importantes da célula, basta escrever o começo

do nome do membro que trará uma lista de todos eles que comecem com as letras digitadas no campo, assim se torna muito mais ágil o cadastramento de célula, além de vincular um membro específico pelo código no banco de dados.

Figura 10 – Tela do sistema eHoly (Cadastro Célula)

## Células

|   |   |  |                      |
|---|---|--|----------------------|
| Nome Célula*                                |   | CEP*                                     |                      |
| <input type="text" value="Eloim2"/>         | <input type="text" value="13480120"/>     | <input type="button" value="Consultar"/> |                      |
| Endereço*                                   |   | Bairro*                                  |                      |
| <input type="text" value="RUA BELIZARIO"/>  |   | <input type="text" value="CENTRO"/>      |                      |
| Estado*                                     | Cidade*                                   | Número*                                  | Complemento          |
| <input type="text" value="SP"/>             | <input type="text" value="LIMEIRA"/>      | <input type="text" value="676"/>         | <input type="text"/> |
| Líder*                                      | Timóteo*                                  |  |                      |
| <input type="text" value="Lucas Nonatto"/>  | <input type="text" value="Eric Nonatto"/> |  |                      |
| Anfitrião*                                  | Supervisor*                               |  |                      |
| <input type="text" value="Sílvia Massaro"/> | <input type="text" value="Lara Nonatto"/> |  |                      |
| Supervisor Área*                            |   |  |                      |
| <input type="text" value="Eric Nonatto"/>   |   |  |                      |

Fonte: Arquivo Pessoal

É nessa funcionalidade que o reconhecimento facial irá ser ingressado, a ideia é que a inteligência artificial detecte a pessoa que está chegando no local por câmeras posicionadas nas entradas do local, esse tema será mais aprofundado no capítulo seis.

### 5.2.1 Lançamento de Presença










Caso o cliente do sistema não possa custear a infraestrutura que o reconhecimento facial precisará, existe essa ferramenta que poderá ser possível fazer os lançamentos de presença de célula ou de eventos se o usuário quiser também. Com uma tela fácil de se entender, o sistema faz ser preenchido o nome da célula e o dia que ele irá lançar as presenças, após clicar no botão pesquisa será mostrado uma lista com todos os membros vinculados a célula, uma vez lançado as presenças clicando na caixa a direita essas informações serão gravadas no banco de dados. Caso o usuário coloque uma data que já tenha sido lançada anteriormente, o sistema traz a lista já preenchida com os membros que tiveram sua presença confirmada, assim é possível consultar ou até lançar novas presenças naquele dia.



Figura 11 – Tela do sistema eHoly (Lançamento Célula)

## Lançamento Presença

Nome Célula  Dia Encontro

| Nome   | Presença                          |
|--|-----------------------------------|
|  Antônio Goncalves Cavalcanti | <input type="checkbox"/> Presente |
|  Bruno Almeida Correia        | <input type="checkbox"/> Presente |
|  Debora Heremam               | <input type="checkbox"/> Presente |
|  Débora Rodrigues             | <input type="checkbox"/> Presente |
|  Diego Araujo Costa           | <input type="checkbox"/> Presente |
|  Eder Gonzales                | <input type="checkbox"/> Presente |
|  Elias Dias                   | <input type="checkbox"/> Presente |
|  Estevan Barros               | <input type="checkbox"/> Presente |
|  Fernanda Paes                | <input type="checkbox"/> Presente |

Fonte: Arquivo Pessoal

Com essa ferramenta é possível manter um controle de quem está comparecendo as reuniões e no final do mês poderá consultar o relatório que será comentado sobre no capítulo de relatório de presença, então assim fazer contato com os membros que não estão indo até o encontro. A ideia principal do sistema é auxiliar os líderes das igrejas a se aproximar cada vez mais dos fiéis de sua denominação e com esse tipo de ferramenta de lançamento de presença fica muito mais fácil se aproximar dos membros que estão mais distantes.

### 5.2.2 Lançamento de Agenda

A ferramenta de lançamento de agenda é uma tela que mostra todas as interações feitas para a agenda, seja ela na criação de eventos ou solicitações de serviço dos membros, marcando o nome dele o dia e a ação que o líder foi fazer.

Figura 12 – Tela do sistema eHoly (Lançamento Agenda)

### Agenda

| Nome                            | Ação        | Data       | Hora     |     |
|---------------------------------|-------------|------------|----------|-----|
| ADILENE MAGNA DIAS RAGAZINI     | Visita      | 25/10/2019 | 10:00:00 | 🔍 📅 |
| ADILENE MAGNA DIAS RAGAZINI     | Visita      | 26/10/2019 | 19:00:00 | 🔍 📅 |
| AILTON DE CASTRO DUARTE         | Visita      | 21/10/2019 | 15:00:00 | 🔍 📅 |
| ALEXANDRE SANTOS DE SOUZA       | Casamento   | 10/10/2019 | 20:00:00 | 🔍 📅 |
| Antônio Goncalves Cavalcanti    | Visita      | 15/08/2019 | 10:00:00 | 🔍 📅 |
| Antônio Goncalves Cavalcanti    | Visita      | 23/08/2019 | 10:00:00 | 🔍 📅 |
| Arthur Reis                     | Aniversário | 05/08/2019 | 10:00:00 | 🔍 📅 |
| CARLOS ALBERTO FILGUEIRA JUNIOR | Aniversário | 15/10/2019 | 09:00:00 | 🔍 📅 |
| CLÁUDIO PINAS COELHO            | Aniversário | 23/10/2019 | 12:00:00 | 🔍 📅 |
| Eliezer Gonçalves de Oliveira   | Visita      | 05/08/2019 | 10:00:00 | 🔍 📅 |

< 1 2 3 >

Fonte: Arquivo Pessoal

É possível perceber que o sistema disponibiliza os eventos serem editados ou apagados, caso haja algum erro de digitação ou até mesmo que o usuário tenha vinculado o membro errado, a ferramenta é muito útil também para caso o usuário queira saber as ações que foram feitas com determinado membro, apenas procurando pelo nome no campo de busca.

### 5.3.1. Relatório de Aniversariantes

Basicamente é a tela que filtra o relatório, o usuário pode escolher por filtrar por determinado mês ou ver todos os meses de uma vez. Uma vez sabendo os aniversariantes do mês o líder da igreja poderá entrar em contato com o membro no dia do aniversário e lhe desejar felicidades o que aproxima ainda mais a igreja e seus fiéis. O relatório retorna o nome, data de aniversário e celular para contato.

### 5.3.2. Relatório de Presença

Neste módulo também é possível filtrar o relatório por mês, mas também pela célula que o usuário deseja ver as presenças. O PDF traz uma lista com os membros da célula e 'X' nos dias que eles compareceram naquele mês

### 5.3.3. Relatório de Membro

Este relatório foi um pedido de um cliente que já está usando o sistema em sua igreja, como é interessante ter as informações dos membros em um arquivo que é possível ser impresso, é colocado nele as informações mais importantes deles, como endereço, nome, contato e número de documentos, além disso a tela de filtro conta com um tipo de relatório que adiciona uma linha para ser assinada pelo membro, caso um dia o líder precise levar essas informações assinadas para sua igreja sede.

Figura 13 – Tela do sistema eHoly (Filtro relatório membro)

Membro

Situação: Todos

Idade Mín.: 1

Idade Máx.: 150

Tipo: Padrão

Estado Civil: Todos

Gerar Relatório

Fonte: Arquivo Pessoal

É possível notar na figura 13 que o filtro também pode ser manipulado a idade mínima e máxima dos membros a serem mostrados no PDF, além do estado civil.

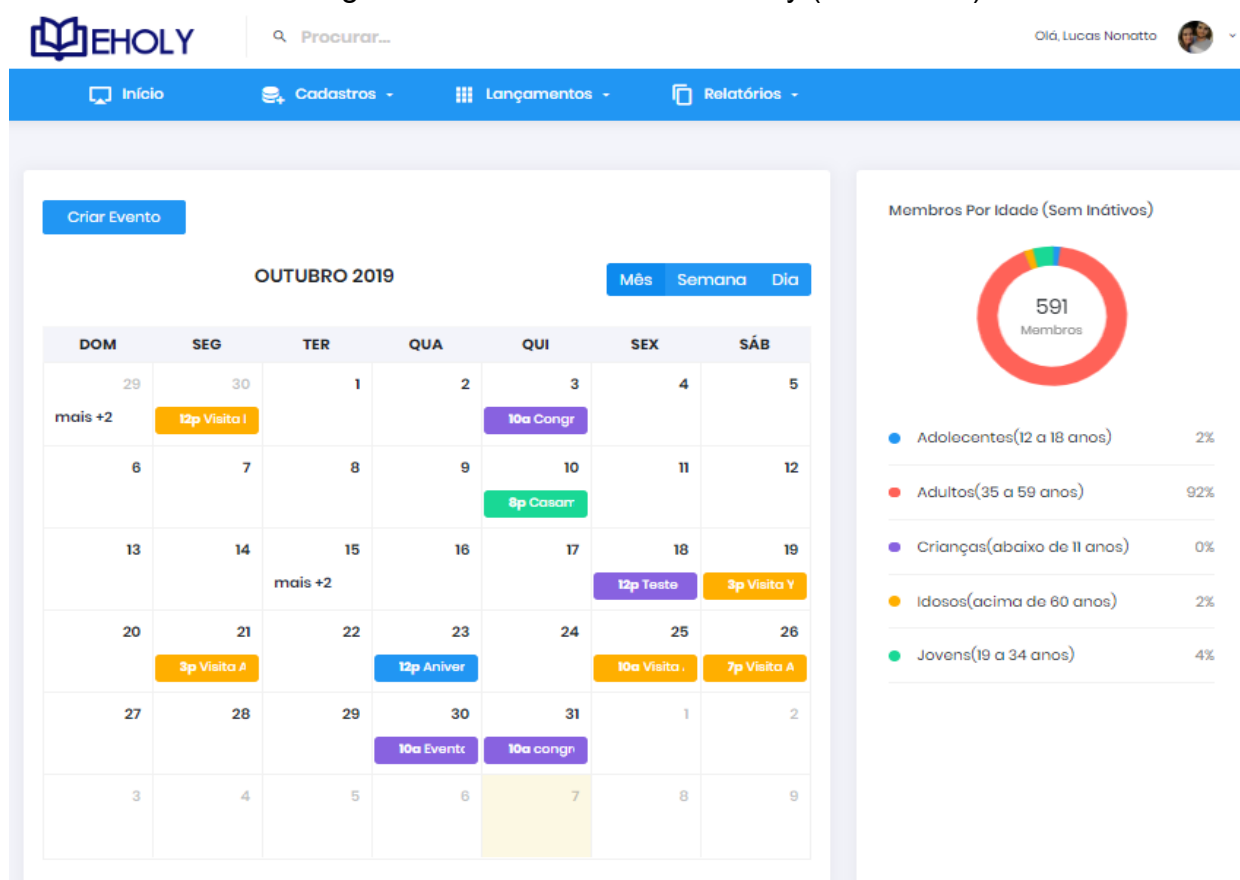
É importante lembrar também que na tela de cadastro de membro é possível imprimir a ficha com todos os dados do membro.

## 5.4. Tela Inicial

A tela mais importante do sistema é provavelmente a tela inicial, pois ela é a que precisa chamar mais atenção do cliente, além de ser agradável aos olhos precisa trazer informações pertinentes no dia a dia, para que o usuário entre no sistema no início do dia e saiba as informações mais importantes do sistema apenas fazendo seu *login*.

Na figura 14 será possível ver que o sistema tem uma tela inicial com todas essas qualidades descritas acima, informações importantes e que seja agradável aos olhos, ou seja, sem poluição visual com informações de mais e que muitas vezes não são úteis.

Figura 14 – Tela do sistema eHoly (Tela inicial)



Fonte: Arquivo Pessoal

Tendo a agenda na primeira tela, o usuário assim que entrar no sistema poderá saber suas tarefas do dia com poucos cliques, destacados com diferentes assim ele poderá saber rapidamente o que ele precisará fazer naquele dia ou na semana.

Mais a direita da tela tem um pequeno gráfico da faixa etária dos membros, assim o líder da igreja poderá saber direcionar os eventos para um público específico e ter uma ideia da quantidade precisa de membros que poderão comparecer nele.

Em cima de todas as telas do sistema existe esse campo de procura que é possível escrever o nome de qualquer membro cadastrado e irá aparecendo os membros em uma lista a baixo do campo conforme o usuário vai escrevendo o nome, mais ou menos como o *Facebook* utiliza no site deles, assim é possível chegar rapidamente ao cadastro do membro,

sem ter que passar pela tela de lista de membro e procurá-lo nela, fica mais fácil que seja possível procurar em qualquer tela do sistema.

## **5.5. Considerações finais sobre o desenvolvimento do sistema web**

Todo o sistema foi desenvolvido utilizando a tecnologia *Bootstrap*, ela é responsável pela responsividade da ferramenta web, ou seja, é possível acessar o sistema de qualquer dispositivo, seja ele um computador com tela grande ou pequena, smartfone ou tablet, não tendo fronteiras para o usuário, somente uma conexão com internet.

Além dessa facilidade para o cliente poder usar o sistema de qualquer dispositivo, foi feito um estudo para que o sistema atendesse da melhor forma possível seus usuários, antes de começar todo o desenvolvimento do sistema, foi pensado em todas as telas, cada *layout* foi pensado e desenvolvido com um acompanhamento de um líder religioso que foi aprovando cada tela e cada funcionalidade do sistema, dando assim mais credibilidade as ferramentas utilizadas e a fluidez da ferramenta web como um todo.

O projeto final foi aprovado por esse líder religioso o qual ficou muito ansioso para utilizar o sistema o quanto antes, até o momento o sistema está sendo usado por ele e mais alguns outros líderes e todos aprovaram o sistema e suas ferramentas, assim mostrando que a tecnologia pode e ajuda muito seja o ramo de atuação que for.

## **6.1. Uso de Python**

Python é uma linguagem de programação interpretada, de código-fonte aberto e com compatibilidade para vários sistemas operacionais. Ser uma linguagem interpretada significa dizer que ao se escrever um programa, este não será compilado, mas sim interpretado por um outro programa que traduzirá para a máquina o que seu programa quer fazer.

O interpretador para Python é interativo, ou seja, é possível executá-lo sem fornecer um código para ele. Ao invés disso, o interpretador disponibilizará uma interface interativa onde é possível inserir os comandos desejados.

Neste projeto a linguagem python está sendo usada para escanear, armazenar e enviar as informações do membro, por exemplo, é executado um arquivo que faz o escaneamento do rosto do indivíduo, em seguida é necessário que a inteligência artificial aprenda que aquele é um rosto válido, e então que as informações sejam enviadas para o banco de dados.

## **6.2. Escaneando**

A parte principal do código que escaneará o rosto da pessoa é a figura a seguir, antes que seja escaneado o rosto do indivíduo o interpretador perguntará o código, deve ser inserido o código correspondente do membro do sistema web, para que na hora de inserir a presença no banco de dados possa vincular as informações do membro com a tabela de presença. Para que isso aconteça com eficácia são tiradas cinquenta fotos da face da pessoa para que tenha uma maior margem de acerto.

Figura 15 – Código Python (Script para escaneamento)

```
while Count < 50:
    ret, img = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    if np.average(gray) > 110:
        faces = face_cascade.detectMultiScale(gray, 1.3, 5)
        for (x, y, w, h) in faces:
            FaceImage = gray[y - int(h / 2): y + int(h * 1.5), x - int(x / 2): x + int(w * 1.5)]
            Img = (NameFind.DetectEyes(FaceImage))
            cv2.putText(gray, "FACE DETECTED", (x+int((w/2)), y-5), cv2.FONT_HERSHEY_DUPLEX, .4, WHITE)
            if Img is not None:
                frame = Img
            else:
                frame = gray[y:y+h, x:x+w]
            cv2.imwrite("dataSet/User." + str(ID) + "." + str(Count) + ".jpg", frame)
            cv2.waitKey(300)
            cv2.imshow("CAPTURED PHOTO", frame)
            Count = Count + 1
cv2.imshow('Face Recognition System Capture Faces', gray)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
```

Fonte: Arquivo Pessoal

As primeiras linhas definem o laço de repetição que no caso é que o código deve ser executado cinquenta vezes, depois o filtro da imagem é modificado para branco e preto para que seja pego mais detalhes do rosto da pessoa, depois é feito um calculo para determinar a posição na tela que está o rosto da pessoa e inserir um quadrado em volta para que o manipulador da ferramenta possa saber que o rosto foi detectado e cada foto é nominada com o código escolhido anteriormente e o número da foto tirada que é baseada no laço de repetição, a foto é salva em uma pasta e depois a variável do contador é acrescido mais um, para que possa chegar até o número de cinquenta fotos.

Lembrando que a figura quinze é apenas uma parte do código que faz escanear o rosto e armazená-lo, existem várias bibliotecas de suporte para que tudo isso aconteça.

### 6.3. Aprendizagem

Para que o reconhecimento facial possa dar certo é preciso que ela seja treinada com todos as faces escaneadas, essa parte quem fica mais responsável é a biblioteca *OpenCV*, primeiro é definido onde deve ser procurado as fotos com os códigos, são definidos dois *arrays*, um com os nomes das fotos e outro com os códigos dos indivíduos, depois é feito o redimensionamento das imagens para ficar no tamanho cento e dez por cem, pois a biblioteca trabalha com esse tamanho, como dito anteriormente, a maior parte do trabalho de inteligência artificial fica com a biblioteca, depois de feita essas preparações, as informações são gravadas em arquivos xml.

Essa é a etapa mais importante do processo de reconhecimento facial, que é ensinar os rostos válidos para a inteligência artificial, a figura dezesseis irá mostrar uma pequena parte do código que foi descrito acima.

Figura 16 – Código Python (Script para aprendizado)

```
path = 'dataSet' # path to the ph
def getImageWithID (path):
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    FaceList = []
    IDs = []
    for imagePath in imagePaths:
        faceImage = Image.open(imagePath).convert('L') # Open image and
        faceImage = faceImage.resize((110,110)) # resize the ima
        faceNP = np.array(faceImage, 'uint8') # convert the im
        ID = int(os.path.split(imagePath)[-1].split('.')[1]) # Retrea
        FaceList.append(faceNP) # Append the Num
        IDs.append(ID) # Append the ID
        cv2.imshow('Training Set', faceNP) # Show the image
        cv2.waitKey(1)
    return np.array(IDs), FaceList # The IDs are co
IDs, FaceList = getImageWithID(path)

# ----- TRAINING THE RECOGNISER -----
print('TRAINING.....')
EigenFace.train(FaceList, IDs) # The recognis
print('EIGEN FACE RECOGNISER COMPLETE...')
EigenFace.save('Recogniser/trainingDataEigan.xml')
print('FILE SAVED..')
FisherFace.train(FaceList, IDs)
print('FISHER FACE RECOGNISER COMPLETE...')
```

Fonte: Arquivo Pessoal

O arquivo varrerá todas as cinquenta fotos de cada membro e aprenderá com elas a determinar qual código deverá ser inserido na hora de mandar as informações para o banco.

#### 6.4. Conexão e envio de dados

O primeiro passo é conexão com o banco utilizando a biblioteca *MySQLdb*, depois será declarado algumas variáveis para que seja possível pegar a data atual do dia, pois utilizaremos ela para inserir junto com as informações pegadas com a câmera, e ajuda na hora de filtrarmos quais membros já foram visto pela câmera naquele dia, na regra de negócio desse sistema é suficiente que seja verificado apenas o dia e não as horas ou minutos, pois cada evento tem em média duas horas, então esse será o tempo que o reconhecimento facial trabalhará, após tudo isso, é iniciado o vídeo da câmera, para que possa ser acompanhado o que a ferramenta está detectando, é declarado a variável de código como zero, pois quando entrar no laço e passar pela área de inserção caso o código seja zero ele não será inserido, é definido que o vídeo será com o filtro branco e preto, e quando for detectado o rosto da pessoa aparecerá um quadrado em volta dele com a informação do código e a confiabilidade que aquele é o rosto vinculado ao código, é definida a variável de inserção do banco e a de seleção, para verificar se o membro que está passando pela imagem da câmera já foi enviado para o banco de dados, com o resultado dessa seleção ficará mais fácil de fazer o filtro de quem já passou pela câmera ou não, caso já não é inserido no banco de dados.

Na hora da inserção de fato, vai ser possível reparar na figura 17 nas últimas linhas do código que a condição para inserção são que a variável de inserção não pode ser vazia, isso é feito pois pode acontecer dela ser vazia até chegar aquele ponto, pois pode ser que nenhum rosto seja detectado e aquela variável não será declarada se isso não acontecer, a outra condição é que o código não pode ser zero, pois se não ficará sendo inserido com valor vazio e a última e possivelmente a mais importante é que só deve ser inserido se a confiança for maior que quarenta por cento. A seguir a figura mostra a parte do código descrita a cima.

Figura 17 – Código Python (Script para inserção)

```

while True:
    NOME = 0
    ret, img = cap.read() # Read the camera object
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # Convert the Camera to gray
    faces = face_cascade.detectMultiScale(gray, 1.3, 5) # Detect the faces and store the posi
    for (x, y, w, h) in faces: # Frames LOCATION X, Y WIDTH, HEIGHT
        # ----- BY CONFIRMING THE EYES ARE INSIDE THE FACE BETTER FACE RECOGNITION IS GAINED -----
        gray_face = cv2.resize((gray[y: y+h, x: x+w]), (110, 110)) # The Face is isolated and cropped
        eyes = eye_cascade.detectMultiScale(gray_face)

        # Executa o comando:
        sql = 'INSERT INTO presenca VALUES (0,29,%s, %s, "pyt")'

        for (ex, ey, ew, eh) in eyes:
            ID, conf = recognise.predict(gray_face) # Determine the ID of the photo
            NAME = NameFind.ID2Name(ID, conf)
            NameFind.DispID(x, y, w, h, NAME, gray)
            NOME = NameFind.ID3Name(ID)

        query = "SELECT codigo_membro FROM presenca WHERE codigo_celula = 29 and data_encontro = '%s'" % (data)
        cursor.execute(query)
        rows = cursor.fetchall()
        result_list = [row for row in rows]
        lista = []
        for row in rows:
            lista.append(str(row[0]))

        if NOME in lista :
            print("Não inserir")
        else:
            if sql != "" and NOME !=0 and conf >= 4000:
                sql_data = (NOME, data)
                cursor.execute(sql, sql_data)
                print('Inserir: ',NOME,conf)

```

Fonte: Arquivo Pessoal

## 6.5. Conclusão Python

A base de tudo isso é a biblioteca OpenCV, que é o coração de toda inteligência artificial do projeto, é ele que faz o escaneamento, armazenamento e reconhecimento das faces, o código escrito foi mais adaptando para a necessidade do sistema.

Essa foi uma parte essencial de todo o projeto, o reconhecimento facial programado por empresas e grupos menores por pessoas está apenas engatinhando, ainda é inviável para que apenas uma pessoa programe uma inteligência dessas e que tenha competitividade comercial, ainda é uma ferramenta com alto custo de desenvolvimento, mas que em um futuro próximo estará tudo mais aberto para os desenvolvedores *freelancers*.

## 7.0 Conclusão

O sistema apresentado neste documento foi feito no intuito de preencher um espaço que está relativamente vazio no mercado, o eHoly que é como o sistema é chamado, tem o objetivo de ajudar toda e qualquer instituição religiosa a manter os dados de seus membros em um lugar seguro e que tenha disponibilidade de consulta vinte e quatro horas por dia, não só consultar, mas também gerar relatórios, marcar eventos vinculados a membros ou não, facilidade na hora de verificar um dado do sistema, com todas essas e muito mais funções que foram descritas neste documento é possível ter um produto de extrema qualidade em um mercado que não existem sistemas com esse grau de complexidade de desenvolvimento mas

que o produto final é algo que agrada o usuário final e que com certeza irá se espalhar ao redor do globo.

A cereja do bolo é o reconhecimento facial implementado na área de confirmação de presença, esta é a maior inovação de todo o sistema, pelas pesquisas feitas, não há igrejas no Brasil com tal ferramenta em uso, além do alto custo de investimento, não há produtos de qualidades que possam fazer tal ação até agora.

O sistema eHoly faz tudo isso e muito mais, além de marcar a presença do membro somente com a confirmação facial é disponibilizado um relatório mensal para que o líder da igreja possa consultar quem está vindo ou não as reuniões. O custo para que isso seja possível são, comprar câmeras apontadas para as entradas do templo, para que tenha um melhor controle de quem está chegando, além disso, no protótipo foi possível usar um minicomputador para fazer os testes com o reconhecimento facial, então sem o investimento necessário para ver o projeto funcionando em grande escala é preciso usar um minicomputador para cada câmera, além desses investimentos, claro que é preciso contratar os serviços do sistema e o valor mensal pode variar de acordo com a quantidade de membros da igreja.

Tendo esses investimentos será possível ter um controle total de sua igreja, com a procura de investimentos para que seja feito em grande escala e consolide de uma vez o sistema como um todo.

## 8.0 Bibliografia

Profissionais ti. Disponível em:

<<https://www.profissionaisiti.com.br/2014/07/o-que-e-computacao-em-nuvem-e-quais-os-principais-tipos/>>. Acesso em: 17 junho. 2019.

Profissionais ti. Disponível em:

<<https://www.profissionaisiti.com.br/2018/05/por-que-sua-empresa-precisa-migrar-para-um-erp-em-nuvem/>>. Acesso em: 17 junho. 2019.

Blog runrun. Disponível em:

<<https://blog.runrun.it/reconhecimento-facial/>>. Acesso em: 15 julho. 2019.

G1. Disponível em:

<<https://g1.globo.com/pop-arte/noticia/2019/05/16/san-francisco-proibe-a-policia-de-usar-reconhecimento-facial.ghtml/>>. Acesso em: 17 julho. 2019.

DeviaMedia. Disponível em:

<<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308/>>. Acesso em: 18 julho. 2019.

Youtube, python. Disponível em:

<<https://www.youtube.com/watch?v=Vnz-TMrwEbM&t=276s/>>. Acesso em: 17 out. 2019.



Diretorio projeto de modelo. Disponível em:

<[https://github.com/igorgbs/python\\_face\\_reco/](https://github.com/igorgbs/python_face_reco/)>. Acesso em: 17 out. 2019.

Documentos *OpenCV*:

<<https://docs.opencv.org/4.1.2/d1/dfb/intro.html>>. Acesso em: 5 agosto. 2019.