

# **Agendamento de Consulta**

**Professores Coordenadores:**

Adiel Seffrin  
Claudinei Dias

**Estudantes:**

Alexandre CAETANO  
Gabriel Helko MEYER  
Jéssica Yohana Otto

# Índice

Introdução.....	3
O Projeto.....	4
Decidido nas primeiras semanas.....	4
Informatização da clínica escola.....	4
Papéis:.....	5
Variáveis e tabelas.....	5
Restrições.....	5
Entrevista com Caio Sestilie.....	7
PHP.....	8
JavaScript.....	9
Configuração.....	10
Instalação.....	10
Iniciando uma Consulta.....	12
Visão do Usuário do Sistema.....	15
Códigos do Projeto.....	19
config.php.....	19
application\config\config.php.....	19
application\config\constants.php.....	19
application\config\database.php.....	20
application\config\routes.php.....	20
application\controller\Appointments.php.....	20
application\controller\Backend_api.php.....	20
application\controller\Backend.php.....	20
application\controller\Consents.php.....	20
application\controller\Console.php.....	20
Implementações futuras.....	21



## Introdução

O Projeto de Extensão: Estúdio de Desenvolvimento de Softwares e Jogos é sobre como podemos facilitar e padronizar o agendamento de consultas de uma Clínica Escola usando um software web com extensão mobile. O projeto ajuda a cadastrar consultas, pacientes, médicos, consultórios, onde pode ter mais de uma sala de atendimento, auxilia no cadastro de triagens, o software também auxilia na consulta a hora marcada, enviando e-mail ou WhatsApp ao paciente, ao prontuário do paciente, ao relatório de cada consulta, administrar as salas de atendimento, quais estão livres e em quais horários estão livres, os médicos disponíveis em cada horário, alterar o horário da consulta do paciente, assim como cancelar a consulta.

## O Projeto

Para atingir os objetivos do projeto, realizou-se estudos dos melhores caminhos para a realização do projeto proposto. O primeiro foi a realização de quadros no Trello e GitHub para a organização do grupo ao longo do projeto.

O segundo foi qual programação escolher, tínhamos inúmeras opções, a escolhida foi o PHP para o Front-End, para o Back-End o JavaScript, para executar o JavaScript no servidor será usado o NodeJs, um programa aberto que possibilita a utilização de JavaScript sem um navegador Web.

O terceiro foi fazer uma pesquisa de mercado e sobre como funcionam outras agendas de consultas para trazer um diferencial software proposto, como fazer para deixar a agenda mais fluida, organizada e com as funcionalidades necessárias esperadas pelos usuários.

Foi decidido utilizar o software de consulta em desenvolvimento Easy!Appointments criado pelo Alex Tselegidis, engenheiro da Grécia, por permitir fazer e customizar as consultas. Foi criado um novo Backend para tratar os dados e fazer a comunicação com o banco de dados.

Endereço do Easy!Appointments: <https://easyappointments.org/>

Endereço do Git do Easy!Appointments: <https://github.com/alextselegidis/easyappointments>

Endereço trello: <https://trello.com/b/g8IRUH3D/agendamento-de-consulta>

Endereço GitHub: <https://github.com/unisociesc/agendamento-de-consulta/projects/1>

## Decidido nas primeiras semanas

### Informatização da clínica escola

- Diversos cursos atendendo no mesmo espaço
- Cadastrar e liberar espaços e horários pelo admin (coordenadores)
- Agendamento dos espaços por ala (consulta):
  - \* Veterinária, Odonto, Nutrição, Psicologia, Fisioterapia, Estética;
- Cadastro do paciente;
- Triagem no cadastro do paciente;
- Agendamento de consulta; (lançamento de emergência)
- Triagem antes da consulta do paciente (pressão e tudo mais);
- Prontuário por paciente (caixa de texto com data e horário);

\* Diagnóstico, tratamento e descrição

- Prescrição
- Histórico de pacientes (relatório)
- Privacidade:

\* O aluno não pode ter acesso ao histórico total do paciente, somente no curso dele.

## **Papéis:**

- \* Coordenador (admin) pode tudo
- \* Preceptor: professor de estágio
- \* Aluno: atendimentos
- \* Secretária: faz os agendamentos dos pacientes
- \* Paciente: cadastro e agendamento. Histórico de prescrições.

## **Variáveis e tabelas**

Cursos: idCurso, Nome

Admin: idAdmin, Nome, Data\_Nasc, Curso\_id

Consulta: idConsulta, Horario, ala, AlunoProfissional\_id, Paciente\_id, Diagnostico, Tratamento, Descricao, Emergencia, Arquivos, Triagem\_id, Ala\_id

Ala\_Medica: idAla, Nome, Espaco

Aluno\_Profissional: idAlunoProfissional, Nome, Data\_Nasc, Especialidade, Historico\_Consulta

Paciente: idPaciente, Nome, Data\_Nasc, Historico\_Consulta, Sintomas

Triagem: idTriagem, AlunoProfissional\_id, Paciente\_id, Horario, Diagnostico

Portuario: idPortuario, Paciente\_id, Relatorio, Arquivos, Especialidade

## **Restrições**

Acesso a informação dos pacientes apenas local.

Não pode haver mais de um agendamento com o mesmo Aluno/Médico no mesmo horário

Deve ser respeitado o limite de consulta por ala

O Aluno/Profissional deve ter informações do paciente apenas da especialidade dele.

Um Relatório deve ser criado a partir da consulta, reunindo as informações do horário, o Aluno/Profissional que atendeu, triagem, diagnóstico, tratamento e descrição, além de imagens de raio-X e afins, esse relatório deve ser montado automaticamente pelo sistema.

Representação de um usuário usando o sistema.

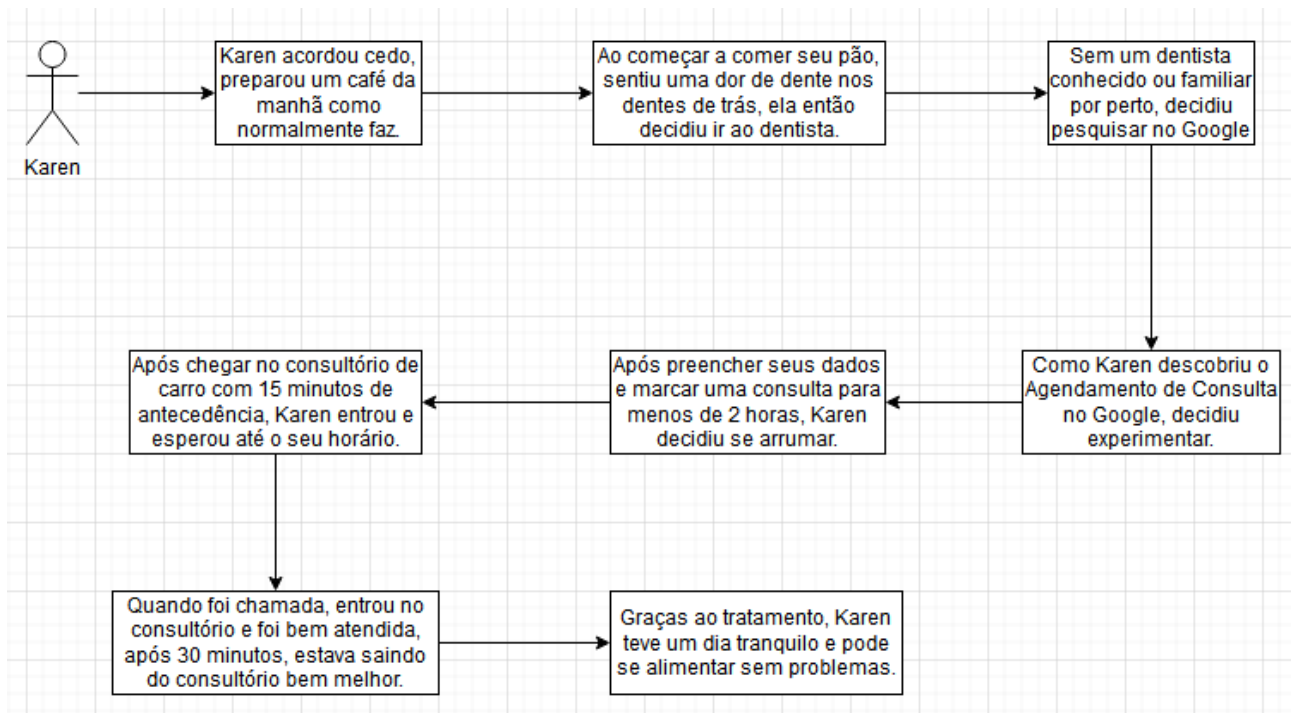


Figura 1: Usuário Karen fictício usando o sistema.

## Entrevista com Caio Sestilie

Entrevista ocorrida dia 19 de maio, às 19:30, sobre o que era esperado do projeto e restrições e funcionalidades que ele teria que possuir.

1. Para cada tipo de atendimento, há mais de uma sala para atender.
2. Para cada curso, pode haver mais de uma pessoa para atender, podendo ter mais de um para o mesmo horário.
3. Ter uma tabela para consultórios, além do tipo de consulta.
4. Nutrição, quatro consultórios, de 1 a 4 profissionais atendendo em certos horários.
5. Cadastrar estagiário, ter seu supervisor no momento, sendo que o estagiário pode ter seus horários de atendimento e seu supervisor pode ver sua agenda, podendo gerar um relatório do supervisor sobre o tempo que ele monitorou o estagiário.
6. Criar uma assinatura eletrônica para o supervisor.
7. Gostou do cartão de agendamento.


8. Confirmação do agendamento por e-mail ou telefone, cópia do agendamento para e-mail.
9. Gera confirmação para secretária quando um usuário confirma a consulta.
10. Estudar utilizar recurso de push.
11. Fazer caminho para a recepcionista fazer o agendamento.
12. Fazer campo periodicidade(períodos, 1 dia, semanas, mês).



# PHP

É um acrônimo recursivo para Hipertext Preprocessor (Pré-processador de Hipertexto), é uma linguagem de desenvolvimento de uso geral, principalmente utilizada no desenvolvimento Web embutida em um HTML.

Suas vantagens são código aberto, suporte avançado, portabilidade do software web convencional para mobile, segurança em relação ao acesso indevido a aplicação, desenvolvimento eficiente de código, estruturas eficazes e funções integradas poderosas.

```
<!DOCTYPE HTML>  Indica para o navegador que o código é HTML
<html>
  <head>
    <title>Exemplo</title>
  </head>
  <body>
    <?php
      echo "Olá, eu sou um script PHP!";
    ?>
  </body>
</html>
```

Pequeno código PHP

Figura 2: Exemplo de código PHP retirado e editado do site [https://www.php.net/manual/pt\\_BR/intro-whatism.php](https://www.php.net/manual/pt_BR/intro-whatism.php)

Para mais informação acessar o [Manual do PHP](#).

Para Download, <https://www.php.net/downloads>

## JavaScript

É uma Linguagem Orientada a Objeto Multiplataforma com Tipagem Dinâmica e Fraca, a parte de Orientada a Objeto significa que toda classe da aplicação é um objeto, com seus métodos, funções e variáveis, que possui o objetivo de transformar a aplicação antes uma estrutura sólida e grande em pequenos módulos conectados por instâncias, isso torna o código mais legível e permite mais fácil manutenção.

Na parte de Multiplataforma, o JavaScript executa dentro de navegadores quando no lado do cliente, isso permite executar da mesma forma seja em Windows, Linux ou MacOS, podendo ter uma versão que é carregada caso a página que está identificando o acesso por um dispositivo mobile, como celular ou tablets.

Sua Tipagem Dinâmica e Fraca permite utilizar variáveis sem definir o seu tipo(String, int, double,...) e alterar entre um tipo ou outro sem precisar de ação do programador, como uma variável que recebia um inteiro pode passar a receber um texto e concatenar(juntar) esse texto com outro texto apenas usando o operador “+”(adição em números ou concatenação em textos).

Suas vantagens é diminuir a codificação para o programador, é muito utilizado no lado do cliente para tratamento de entradas no lado do Cliente, para diminuir o trabalho no servidor, já no servidor, permite uma comunicação fluída entre os objetos da aplicação com o Banco de Dados.

<pre>var pessoa1 = {   nome: 'Chris',   saudacao: function() {     alert('Oi! Meu nome é ' + this.nome + '.');   } }</pre>	Um Objeto Pessoa.
<pre>var pessoa2 = {   nome: 'Brian',   saudacao: function() {     alert('Oi! Meu nome é ' + this.nome + '.');   } }</pre>	Outro Objeto Pessoa.

Figura 3: Exemplo de Código de dois Objetos utilizando a mesma variável no mesmo código mas guardando valores diferentes, retirado e editado do site <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/Basics>

Para mais informações, acessar o [Site Oficial](#).

## Configuração

Para clonar a aplicação e executar ela na sua máquina, será necessário o [GIT](#), [Node.js](#) (ao qual já vem com [npm](#)) e [Composer](#), após a instalação de todos, no GIT, insira:

```
# Clone this repository
$ git clone https://github.com/alextselegidis/easyappointments.git

# Go into the repository
$ cd easyappointments

# Install dependencies
$ npm install && composer install

# Start the file watcher
$ npm start
```

*Figura 4: Comandos a ser digitados na tela do Git*

No primeiro comando, clona o repositório, no segundo coloca o prompt no repositório, no terceiro instala npm e composer nele, no quarto comando, inicia o npm.

Obs.: Caso esteja usando um terminal de comando Linux no Windows, de uma olhada [nesse tutorial](#) para utilizar a aplicação gráfica do Linux no Windows, você também pode usar o Node para executar os comandos.

## Instalação

Você vai precisar seguir alguns passos para a instalação da aplicação no seu servidor.

- Tenha certeza de o servidor a ser usado tenha Apache/Nginx, PHP e MySQL instalados, uma sugestão é o [Vertrigo](#).
- Crie um novo banco de dados ou utilize um existente.
- Copie o código fonte do “easyappointments” para o servidor.
- Tenha certeza que é possível escrever/alterar no servidor.
- Renomeie o “config-sample.php” que está na aplicação para “config.php”, depois atualize o conteúdo em relação ao seu banco de dados.

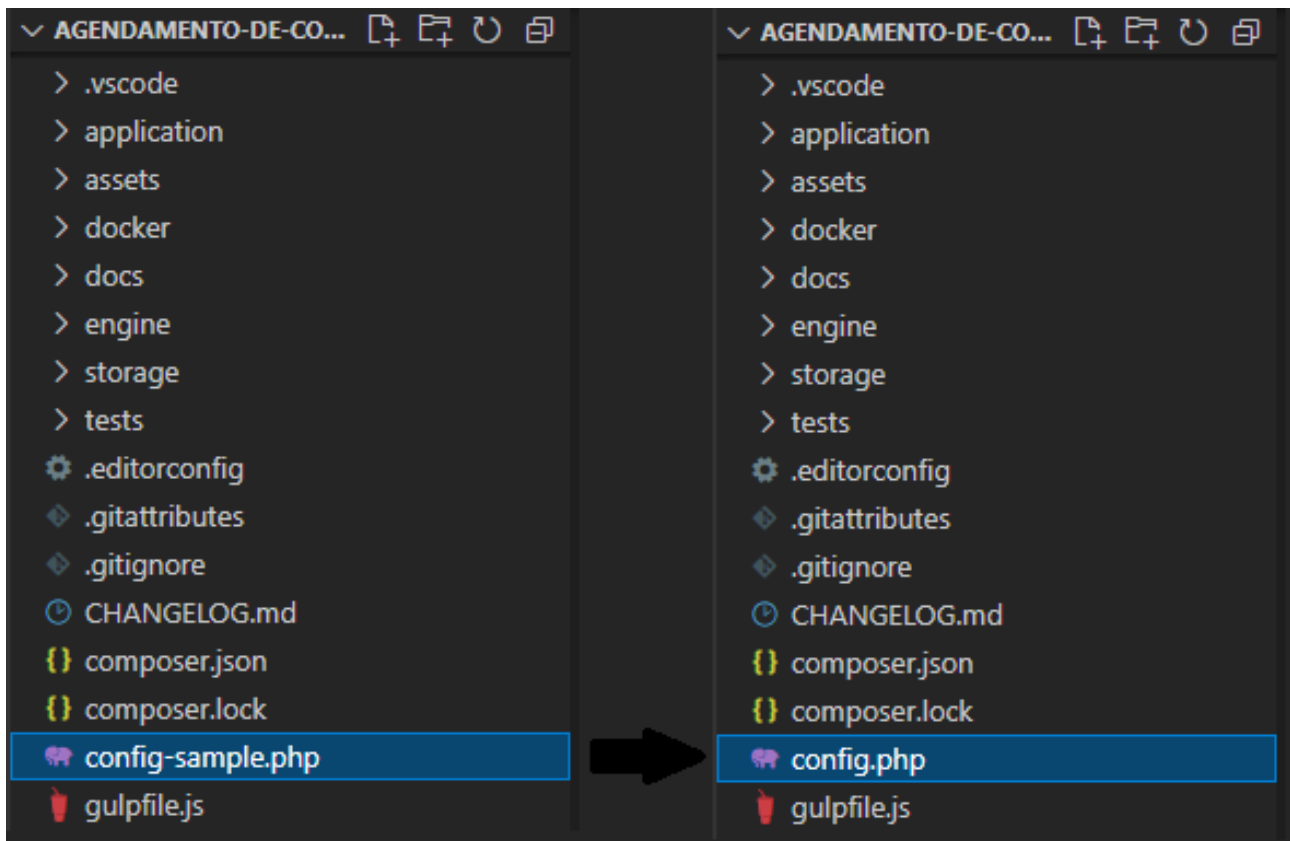


Figura 5: No Microsoft Visual Studio, no Explorer de arquivos, o config que deve ser alterado

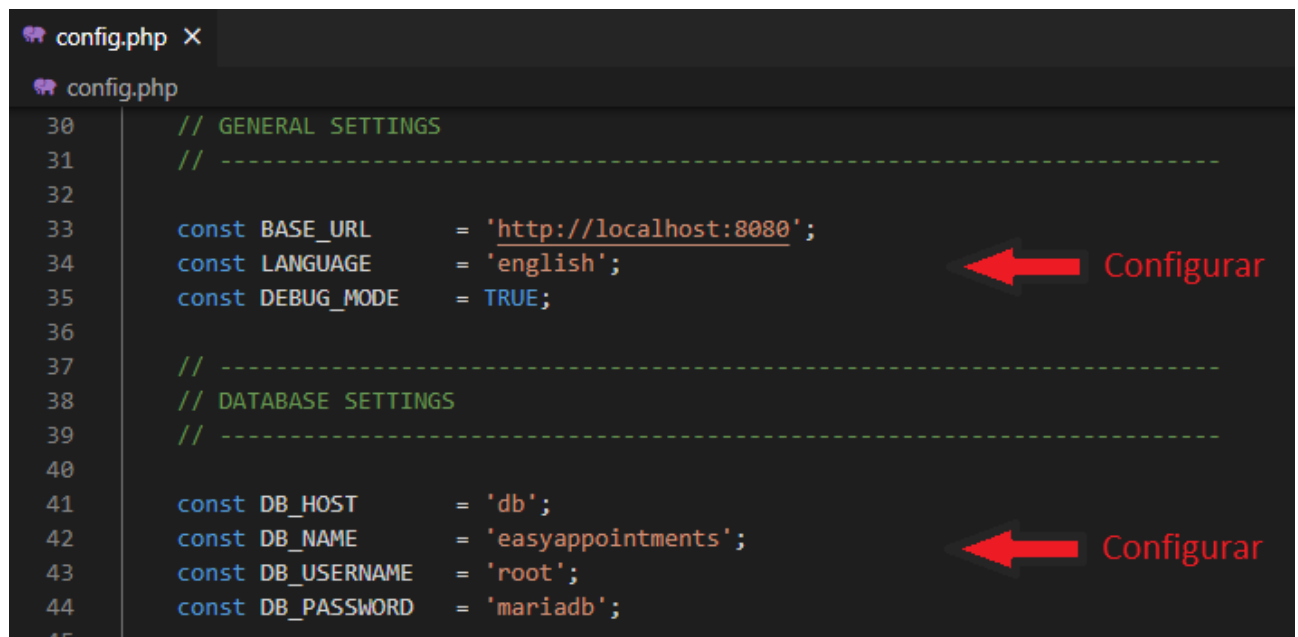


Figura 6: Partes que devem ser alteradas para a conexão com o Banco de Dados.

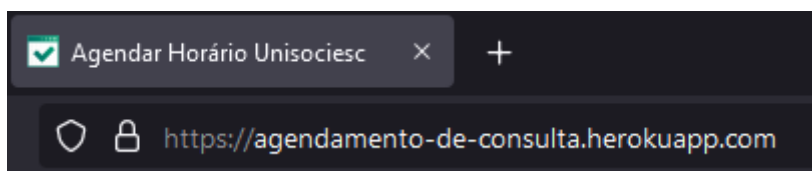
- Abra o seu navegador na URL do Easy!Appointments e siga as instruções de instalação.

Com isso, você já pode executar a aplicação. Você pode baixar a última versão do Easy! Appointments [aqui](#), caso tenha dificuldade na instalação ou utilização, acesse o grupo [Oficial de Supote](#), caso encontre um erro e queira relatar, acesse a [página de problemas](#).

## Iniciando uma Consulta

Aqui será demonstrado o passo a passo para marcar uma consulta.

1 – Entre no site [Agendamento de Consulta](https://agendamento-de-consulta.herokuapp.com).



*Figura 7: Site do Agendamento de Consulta, por tempo limitado.*

2 – Na tela de início, selecione o serviço que deseja e o atendente que atenderá você. Depois clique em Seguinte.

**Serviço**

Consulta Direito ▾

**Atendente**

- Qualquer Atendente - ▾

**Consulta Direito**  
[Duração 30 Minutos]

Seguinte >

*Figura 8: Seleção de serviços e atendentes na tela inicial.*

3 – Selecione a data que deseja marcar a consulta no calendário, depois, selecione o fuso horário do seu estado, um menu aparecerá com horas embaixo do fuso horário, selecione um horário disponível para ser marcado sua consulta. Depois aperte em Seguinte.

The interface shows a calendar for June 2021 on the left, with the 30th highlighted. To the right, a dropdown menu for 'Fuso horário' is set to 'Brasília (-3:00)'. Below this, a list of available times is shown: 09:00 (highlighted in red), 09:15, 09:30, 09:45, and 10:00. At the bottom of the form are two buttons: 'Voltar' (Back) and 'Seguinte' (Next).

Figura 9: Seleção de data e horário.

4 – Preencha suas informações pessoais para registrar a consulta.

The form consists of two columns of input fields. The left column contains: 'Primeiro Nome \*', 'Ultimo Nome \*', 'Email \*', and 'Telefone \*'. The right column contains: 'Endereço', 'Cidade', 'CEP', and 'Observações.'. Each field is a simple text input box. At the bottom of the form are two buttons: 'Voltar' (Back) and 'Seguinte' (Next).

Figura 10: Ficha simples de dados dos clientes.

5 – Confirme suas informações e aperte em Confirmar para marcar sua consulta.

Agenda	Cliente
Serviço: Consulta Direito	Cliente: [REDACTED]
Atendente: - Qualquer Atendente -	Telefone: [REDACTED]
Início: 30/06/2021 09:00	Email: [REDACTED]
Fuso horário: Brasília (-3:00)	Endereço: [REDACTED]
	Cidade: [REDACTED]
	CEP: [REDACTED]

[< Voltar](#)[Comfirmar](#)

Figura 11: Tela para confirmar informações digitadas anteriormente.

6 – Você receberá a confirmação da consulta marcada e um aviso de e-mail enviado, contendo detalhes da sua consulta. Ao clicar em ir para página de reserva, você irá para a página inicial novamente.

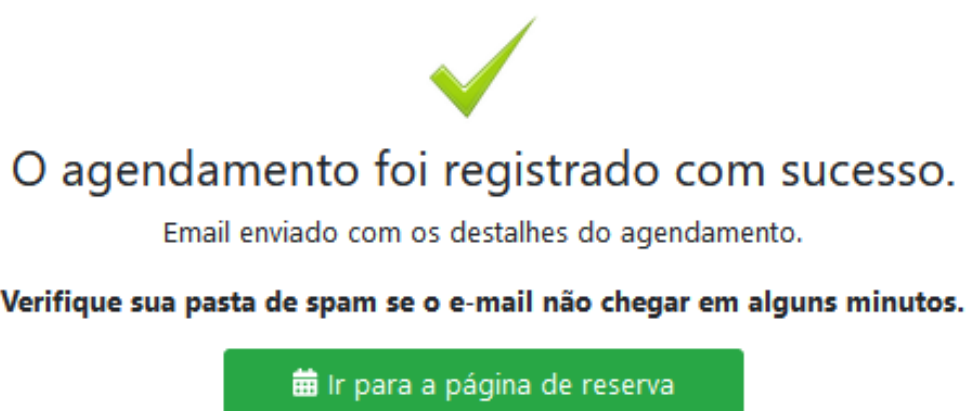


Figura 12: Mensagem de consulta marcada com sucesso.

## Visão do Usuário do Sistema

Para entrar no sistema, clique no botão Login, na parte inferior a direita da página, um pouco abaixo do botão seguinte.



*Figura 13: Botão Login na página inicial.*

Na seção de login, digite o nome do usuário(aqui estamos usando o e-mail) e sua senha para entrar. Caso tenha esquecido sua senha, clique em Esqueci minha senha e siga as instruções.

### Seção de backend

Bem-Vindo! Precisa entrar para visualizar todo o conteúdo

Nome de usuário

Senha



[Esqueceu a senha?](#) | [Portuguese-br](#)

*Figura 14: Menu de Login.*

Após o login, você estará nessa tela principal do sistema, onde é possível visualizar o calendário de cada aluno/profissional cadastrado, ao clicar em cima de uma consulta marcada, um pequeno menu abrirá mostrando informações mais detalhadas da consulta, também terá a opção de fechar, excluir ou editar a consulta, você poderá alterar o horário da consulta apenas arrastando ela e colocando em outro horário vago.

No campo de seleção à esquerda abaixo do logo da UniSociesc você pode escolher o aluno ou serviço que deseja visualizar o calendário. Há direita, no campo de seleção é possível adicionar um horário onde o aluno/profissional não poderá atender na opção Exceção do plano de trabalho, ou poderá marcar um ou mais dias onde estará indisponível, na opção Indisponível. O botão ao lado da caixa de seleção Agenda é um botão para atualizar os agendamentos do calendário, carregando os novos agendamentos se existirem e o próximo botão, o botão mais à direita, é o botão para alterar a



visualização do calendário, alterando entre o calendário de só um aluno/profissional para a visão de todos, permitindo alterar aparência do calendário para a de uma lista, listando seus compromissos por horário, ele também permite listar até três dias à frente dessa maneira.

No menu acima, você pode selecionar o Calendário, que inicia por padrão, Clientes, onde poderá consultar, cadastrar, alterar e deletar as informações privadas de cada cliente, Serviços, onde poderá consultar, cadastrar, alterar e deletar serviços, Usuários, onde as pessoas que vão trabalhar são cadastradas, alteradas ou deletadas, também a opção Definições, onde a regra de negócio do site é alterada, essa opção é só para desenvolvedores, por último, Sair, que encerra a sessão do usuário no sistema.

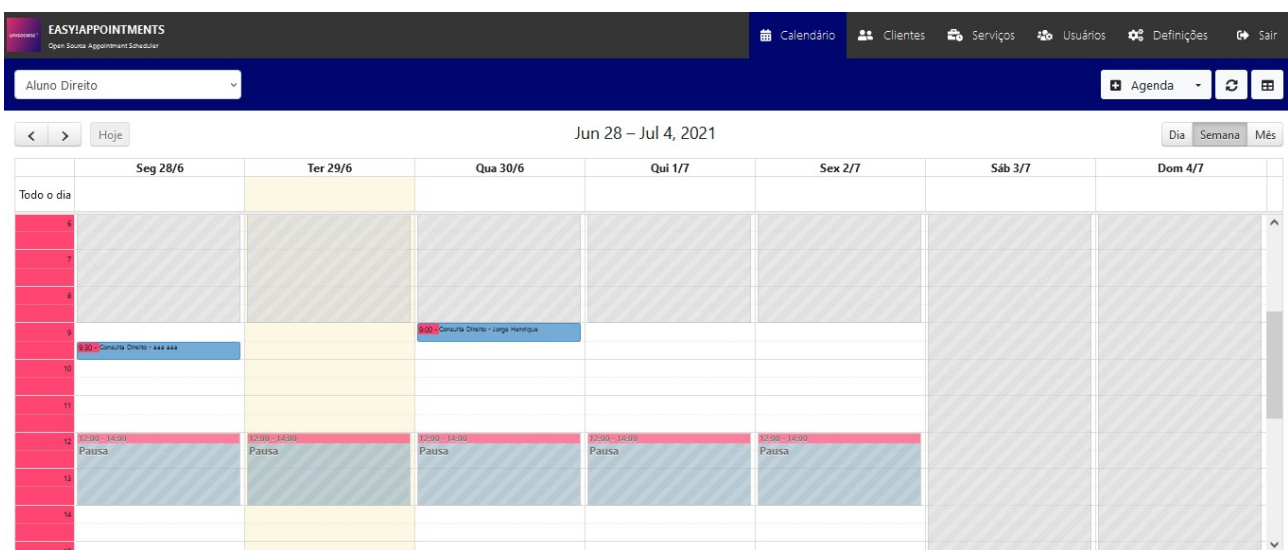


Figura 15: Página inicial após o login.

O menu de clientes e suas funções.

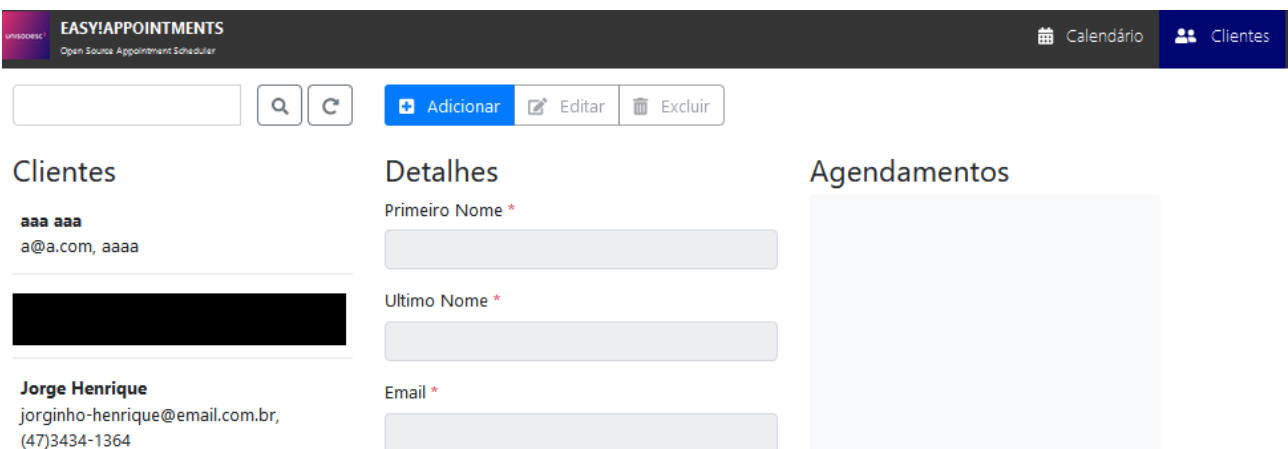


Figura 16: Menu de Clientes.

O menu de serviços e suas funções.

**EASY!APPOINTMENTS**  
Open Source Appointment Scheduler

Calendário Clientes **Serviços**

Serviços Categorias

Adicionar Editar Excluir

**Serviços**

**Consulta Odontologica**  
30 min - 0.00

**Consulta Direito**  
30 min - 0.00

**Detalhes**

Nome \*

Duração (minutos) \*

Valor \*

Figura 17: Menu de Serviços.

O menu de usuários e suas funções.

**EASY!APPOINTMENTS**  
Open Source Appointment Scheduler

Calendário Clientes Serviços **Usuários**

Atendentes Secretários Administradores

Adicionar Editar Excluir Detalhes Plano de trabalhos

**Atendentes**

**Aluno Odontologia**  
jane@example.org, +1 (000) 000-0000

**Aluno Direito**  
sda@asda.com, 464

**Detalhes**

Primeiro Nome \*

Ultimo Nome \*

Email \*

Telefone \*

Nome de usuário \*

Senha \*

Repetir Senha \*

Calendário \*

Default

Figura 18: Menu de Usuários.

Menu de definições, onde administradores poderão alterar opções do programa, alterar regra de negócio, como horários ativos. Em Conteúdos Legais você pode definir a utilização de cookies ou políticas de privacidades. Já em Usuário Atual, você poderá ter acesso às suas informações, que são especiais por você ser um administrador. Por último, está um guia de informações do Easy!Appointment.

**EASY!APPOINTMENTS**  
Open Source Appointment Scheduler

Calendário Clientes Serviços Usuários **Definições**

**Geral** Lógica de negócio Conteúdos Legais Usuário Atual Sobre o Easy!Appointments

Definições Gerais

**Nome da empresa \***  
  
Nome da empresa será exibido em todos os lugares no sistema (obrigatório).

**Email da empresa \***  
  
Este será o endereço de email da empresa. Ele será usado como o remetente e o endereço de resposta dos emails do sistema (obrigatório).

**Endereço web da empresa \***

**Google Analytics ID**  
  
Adicione o seu ID do Google Analytics para ser incluído na página de agendamentos.

**API Token**  
  
Defina um token secreto para permitir a autenticação baseada em tokens do Easy!Appointments API.

☒ **Notificações de Clientes**  
Define se o cliente receberá ou não, notificações por email

*Figura 19: Menu de Definições.*

## Códigos do Projeto

Aqui será demonstrado alguns códigos do projetos e suas funcionalidades

### config.php

```
29 // -----  
30 // GENERAL SETTINGS  
31 // -----  
32  
33 const BASE_URL      = 'http://localhost:8080';  
34 const LANGUAGE      = 'english';  
35 const DEBUG_MODE    = TRUE;
```

Figura 20: Configurações Gerais no Config.php

Defini as configurações gerais da aplicação, na BASE\_URL o endereço e porta da aplicação, na LANGUAGE define o idioma da aplicação e em DEBUG\_MODE, permite obter maiores resultados da aplicação, principalmente retornos de CRUD.

```
37 // -----  
38 // DATABASE SETTINGS  
39 // -----  
40  
41 const DB_HOST        = 'phpmyadmin';  
42 const DB_NAME        = 'easyappointments';  
43 const DB_USERNAME    = 'root';  
44 const DB_PASSWORD    = 'vertrigo';
```

Figura 21: Configurações de Banco de Dados no Config.php

Configurações do banco de dados, DB\_HOST indica o software de banco utilizado, o DB\_NAME indica o banco de dados onde está às tabelas utilizadas, DB\_USERNAME é o login do usuário do banco de dados e DB\_PASSWORD sua senha, essa configuração permite o software acessar o banco de dados para fazer os CRUD.

### application\config\config.php

Arquivo onde é feita a configuração do software, linguagens a utilizar, tipo de caracteres aceito na url, diretório de erros, formato da data, etc.

### application\config\constants.php

Começando na linha 48, define constantes que serão utilizadas ao usar dados.

## **application\config\database.php**

Configura a conexão com o banco de dados(hostname, username, password, database), qual o tipo de banco que será utilizado(driver), o tipos de caracteres utilizados pelo banco(collat), entre outros. Os “Config:” que aparecem na imagem acima estão pegando a informação do config.php que está na pasta principal do projeto.

## **application\config\routes.php**

Definição das rotas que serão usadas ao acessar recursos no aplicativo.

## **application\controller\Appointments.php**

Onde é feito o controle, utilizando rotas para outros módulos, verificações de erros, carregamento de variáveis com valores que serão utilizados para consultar, marcar consultas ou deletar consultas.

## **application\controller\Backend\_api.php**

Nessa classe, o calendário é montado e tratado, com métodos para fazer ou chamar as consultas e mostrá-las em tela, inserção de filtros de pesquisa, tratamento de e-mail's, tratamento de usuários, consultas, médicos, etc.

## **application\controller\Backend.php**

Inicia as variáveis que serão usadas na aplicação e define seus valores de acordo com valores pré-definidos, permite alteração de acesso do usuário pelo administrador, quais variáveis serão acessadas e mostradas em cada tipo de acesso na aplicação e configurações adicionais para administradores.

## **application\controller\Consents.php**

Salva o consentimento do usuário usando seu IP e id em um arquivo JSON.

## **application\controller\Console.php**

Configura comando para utilizar o PHP console, possui a instalação, os primeiros dados a serem colocados no banco, backups e ativação ou desativação de sincronização do Google.

## Implementações futuras

Aqui estão algumas sugestões de implementações que podem contribuir para o desenvolvimento e polimento do projeto.

1. Colocar relatórios necessários para os médicos(diagnóstico do paciente, lista de laudos médicos do paciente, entre outros).
2. Atualizar cronogramas no GitHub de tarefas feitas e à fazer.
3. Montar conexão com o banco de dados para salvar dados.
4. Montar um banco mais consistente em relação às informações mostradas ao usuário.
5. Limpar o código de comentários desnecessários e códigos já não mais utilizados.
6. Corrigir o nome de variáveis no código para variáveis mais condizentes com a proposta.
7. Verificar a opção de remover o nome Easy!Appointments das telas de acesso sem ferir os direitos da licença.
8. Reunião semanal da equipe para conversar sobre o projeto, com dia e horário onde todos os membros possam estar presentes(uma sugestão da nossa equipe).