

UNIFEOB  
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO BASTOS  
ESCOLA DE NEGÓCIOS  
**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**  
**GESTÃO DE TECNOLOGIA DA INFORMAÇÃO**

**PROJETO INTEGRADO**  
**IGNITE CALL – AGENDAMENTO DESCOMPLICADO**

SÃO JOÃO DA BOA VISTA, SP  
JUNHO 2023

UNIFEOB  
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO BASTOS  
ESCOLA DE NEGÓCIOS  
**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**  
**GESTÃO DE TECNOLOGIA DA INFORMAÇÃO**

**PROJETO INTEGRADO**  
**IGNITE CALL – AGENDAMENTO DESCOMPLICADO**

MÓDULO DESENVOLVIMENTO DESKTOP  
Ferramentas de Desenvolvimento Web – Prof. Nivaldo de Andrade  
Sistemas Operacionais – Prof. Rodrigo Marudi de Oliveir  
Projeto de Desenvolvimento Web – Prof. Nivaldo de Andrade

Estudantes:

Eduardo Coelho, RA 1012022100885

SÃO JOÃO DA BOA VISTA, SP  
JUNHO, 2023

## Sumário

1 INTRODUÇÃO.....	4
2. OBJETIVOS.....	5
3 PROJETO DE CONSULTORIA EMPRESARIAL.....	6
3.1 Requisitos funcionais.....	6
3.2 FERRAMENTAS PARA DESENVOLVIMENTO WEB.....	7
3.2.1 Projetando a Ferramenta.....	7
3.2.2 Projeto de interface com o usuário.....	9
3.2.3 Linguagem de Desenvolvimento.....	10
3.3 SISTEMAS OPERACIONAIS.....	14
3.3.1 COMPONENTES DE SISTEMAS OPERACIONAIS.....	14
3.3.2 GERENCIAMENTO E FUNCIONALIDADES DO SISTEMA OPERACIONAL.....	14
3.3.3 GERENCIAMENTO DE HARDWARE PELO SISTEMA OPERACIONAL.....	16
4 CONCLUSÃO.....	18
REFERÊNCIAS.....	19

# 1 INTRODUÇÃO

Devido a uma observação na empresa de um conhecido, um escritório de contabilidade, foi notado que grande parte dos funcionários infelizmente acabavam esquecendo dos seus afazeres, como por exemplo: reuniões, prazos, metas, entre outros. Gerando uma rotina de trabalho totalmente desorganizada. Foi quando surgiu a oportunidade de estar automatizando e dando mais agilidade nos compromissos do dia a dia de trabalho. Para a implementação do sistema ira ser utilizada as tecnologias Typescript, o framework React, MySQL como banco de dados e a a API do Google.

## **2. OBJETIVOS**

Este projeto tem como objetivo ajudar no gerenciamento da agenda de trabalho dos colaboradores do escritório. Neste aplicativo, o usuário poderá realizar login no site utilizando a sua conta do google, permitindo que receba compromissos da agenda usando a API do Google, um projeto semelhante ao Calendly.

### **3 PROJETO DE CONSULTORIA EMPRESARIAL**

O sistema será iniciado com uma prévia entrevista ao proprietário e os futuros usuários do software, onde serão levantados todos os requisitos, necessidades, relatórios, cadastros necessários. Para isso, será utilizada a técnica de orientado a objetos. O desenvolvimento será dividido em partes de necessidades, onde será apresentado aos interessados o andamento do desenvolvimento, evitando assim transtornos e erros no final do projeto.

Nessa etapa, obtêm-se informações com os stakeholders (usuários finais e pessoas da organização que serão afetadas pelo software) sobre o que ou não o software deverá fazer. As atividades relacionadas ao processo de análise de requisitos são: descoberta, classificação e organização, priorização e negociação, especificação. Para a análise de requisitos, geralmente são realizadas entrevistas com os stakeholders, que, segundo Sommerville, podem ser do tipo fechada, em que são definidas as perguntas da entrevista, e do tipo aberta, onde não são perguntas predefinidas, mas é feita uma análise com os stakeholders para uma melhor compreensão de suas necessidades. É nessa etapa também que são definidos os requisitos funcionais e não funcionais. Após o levantamento dos requisitos vem a fase da análise desses requisitos, para avaliar se todas as necessidades dos usuários foram entendidas corretamente e determinar se algo precisa ser melhorado.

#### **3.1 Requisitos funcionais**

Os requisitos funcionais descrevem o que este deverá fazer, ou seja, as funcionalidades do software, de forma geral ou específica, de acordo com o tipo de software a ser desenvolvido. É importante que, ao especificar os requisitos, todos os serviços e funções solicitados pelo cliente sejam definidos e essas definições sejam coerentes. Dessa forma, o software poderá ser entregue mais rápido, sem necessidade de rever o desenvolvimento por causa de requisitos mal definidos ou inconsistentes

## 3.2 FERRAMENTAS PARA DESENVOLVIMENTO WEB

**Linguagens de marcação:** Linguagens como HTML (Hypertext Markup Language) são usadas para estruturar e organizar o conteúdo de uma página web. O HTML define os elementos e tags que compõem a estrutura básica de uma página.

**Folhas de estilo:** Cascading Style Sheets (CSS) é uma linguagem que define a aparência e o layout dos elementos HTML em uma página. Com o CSS, é possível controlar cores, fontes, tamanhos, margens, posicionamento e outros aspectos visuais da página.

**Linguagem de programação:** JavaScript essa linguagem permite adicionar interatividade e funcionalidade às páginas web, permitindo a manipulação de elementos, a validação de formulários, a comunicação com servidores e a criação de lógica de negócios.

**Framework:** React é uma estrutura e conjunto de ferramentas para acelerar o desenvolvimento web. Ele oferece recursos pré-construídos, como componentes reutilizáveis, roteamento, gerenciamento de estado e integração com APIs, facilitando a criação de aplicativos web complexos e escaláveis.

**Banco de dados:** Para armazenar e recuperar informações foi utilizado um sistema de gerenciamento de banco de dados, como MySQL. Esse sistema permitem a criação, a leitura, a atualização e a exclusão de dados de forma eficiente e segura.

**Controle de versão:** O controle de versão, geralmente feito com o uso de sistemas como Git, permite que equipes de desenvolvimento colaborem de forma eficiente no desenvolvimento de uma aplicação web. Ele permite o rastreamento de alterações, o gerenciamento de ramificações (branches) e a fusão (merge) de código, além de facilitar o trabalho em equipe e a reversão de alterações, se necessário.

### 3.2.1 Projetando a Ferramenta

**Definindo uma aplicação web:** Uma aplicação web é um software que é executado em um navegador web e é acessado por meio da internet. Diferente de aplicativos tradicionais que requerem instalação em dispositivos específicos, as aplicações web podem ser acessadas de qualquer lugar e em qualquer dispositivo com um navegador e conexão com a internet. Elas são desenvolvidas utilizando tecnologias web, como HTML, CSS e JavaScript, e podem fornecer funcionalidades diversas, desde sites simples até aplicações complexas.

**Principais fases de desenvolvimento de uma aplicação web:** O desenvolvimento de uma aplicação web geralmente passa por várias fases, que podem variar dependendo da metodologia adotada e das necessidades específicas do projeto. Aqui estão as principais fases:

**Análise de requisitos:** Nesta fase, é feita a compreensão dos objetivos da aplicação web, identificando os requisitos funcionais e não funcionais. Isso envolve a coleta de informações sobre as necessidades do cliente, o público-alvo, os recursos desejados e as restrições técnicas.

**Design e prototipagem:** Com base nos requisitos coletados, é criado um projeto de design que define a estrutura da aplicação, a navegação, o layout e a aparência visual. A prototipagem pode ser utilizada para validar o design antes do desenvolvimento.

**Desenvolvimento:** Nesta fase, a aplicação web é construída, envolvendo a implementação do código HTML, CSS e JavaScript, bem como a integração com bancos de dados e serviços externos, se necessário. Frameworks e bibliotecas podem ser utilizados para agilizar o desenvolvimento.

**Testes:** Os testes são essenciais para garantir a qualidade e o funcionamento correto da aplicação web. Isso pode incluir testes de funcionalidade, testes de desempenho, testes de segurança e testes de usabilidade. Correções e ajustes são feitos com base nos resultados dos testes.

**Implantação:** Após os testes e ajustes finais, a aplicação web é implantada em um ambiente de produção, geralmente em um servidor web. Isso envolve a configuração do ambiente de hospedagem, a instalação de dependências e a garantia de que a aplicação esteja pronta para ser acessada pelos usuários.

**Manutenção e atualizações:** Após a implantação, a aplicação web requer manutenção contínua para corrigir bugs, adicionar novos recursos, otimizar o desempenho e lidar com possíveis problemas de segurança. Atualizações e melhorias devem ser feitas regularmente para acompanhar as necessidades do usuário e as mudanças tecnológicas.

#### **Principais erros e problemas no desenvolvimento de uma aplicação web:**

- Falta de planejamento adequado e análise de requisitos, resultando em um produto que não atende às necessidades do usuário.
- Má arquitetura e design, resultando em uma aplicação web difícil de manter, escalar ou modificar.
- Má usabilidade e experiência do usuário, tornando a aplicação difícil de usar e frustrante para os usuários.
- Problemas de desempenho, como carregamento lento de páginas, tempo de resposta

### 3.2.2 Projeto de interface com o usuário

Interação Humano-Computador (IHC) em ambientes web refere-se ao estudo e design das interações entre os usuários e as aplicações web, com o objetivo de melhorar a usabilidade, a eficiência e a satisfação do usuário. A IHC se concentra em entender as necessidades, habilidades e características dos usuários, bem como nos aspectos técnicos e de design das aplicações web.

Princípios e boas práticas de IHC em ambientes web:

1. **Feedback e visibilidade:** É importante fornecer feedback claro e imediato aos usuários sobre suas ações. Isso pode incluir mensagens de confirmação, indicadores de progresso e atualizações em tempo real. Além disso, a visibilidade das opções disponíveis e do estado atual do sistema é fundamental para que os usuários entendam o que está acontecendo.
2. **Simplicidade e consistência:** As interfaces web devem ser simples e intuitivas, com fluxos de trabalho claros e concisos. Manter uma consistência visual e de interação dentro da aplicação ajuda os usuários a se sentirem mais confortáveis e a encontrar informações e funcionalidades de forma mais eficiente.
3. **Facilidade de aprendizado:** As aplicações web devem ser projetadas para serem fáceis de aprender e usar, mesmo por usuários iniciantes. Isso pode ser alcançado por meio de elementos visuais claros, instruções simples e suporte contextual, como dicas de ferramentas e ajuda incorporada.
4. **Flexibilidade e controle do usuário:** Oferecer opções de personalização e permitir que os usuários tenham controle sobre a aplicação web aumenta a sensação de controle e satisfação. Isso pode incluir preferências de visualização, configurações de notificação e capacidade de desfazer ações.
5. **Design responsivo:** Com o aumento do uso de dispositivos móveis, é essencial que as aplicações web sejam responsivas, adaptando-se a diferentes tamanhos de tela e oferecendo uma experiência de usuário consistente em todos os dispositivos.
6. **Acessibilidade:** As aplicações web devem ser acessíveis a todos os usuários, independentemente de suas habilidades ou limitações. Isso envolve o uso de práticas como a utilização adequada de marcadores semânticos, fornecimento de alternativas textuais para elementos visuais e suporte a tecnologias assistivas.
7. **Testes e avaliação contínua:** É importante realizar testes de usabilidade e coletar feedback dos usuários para identificar problemas e oportunidades de melhoria. A avaliação contínua da interface ajuda a otimizar a experiência do usuário e a corrigir quaisquer problemas que possam surgir.

Esses princípios e boas práticas de IHC fornecem diretrizes para projetar e desenvolver aplicações web que sejam intuitivas, eficientes e agradáveis de usar. É importante considerar o contexto dos usuários e adaptar essas práticas de acordo com as necessidades específicas da aplicação web e de seu público-alvo.

### **3.2.3 Linguagem de Desenvolvimento**

Para realizar o projeto a linguagem de programação escolhida foi o TypeScript que é uma linguagem de programação de código aberto desenvolvida pela Microsoft. É um superconjunto sintático estrito de JavaScript e adiciona tipagem estática opcional à linguagem. Também foi utilizado o react-js como framework, que é uma estrutura e conjunto de ferramentas para acelerar o desenvolvimento web. Ele oferece recursos pré-construídos, como componentes reutilizáveis, roteamento, gerenciamento de estado e integração com APIs, facilitando a criação de aplicativos web complexos e escaláveis.

Para a página home foi utilizado o seguinte código:

```
1 import { Heading, Text } from '@ignite-ui/react'
2 import Image from 'next/image'
3 import { Container, Hero, Preview } from './styles'
4 import previewImage from '../assets/appPreview.png'
5 import { ClaimUsernameForm } from './components/ClaimUsernameForm'
6 import { NextSeo } from 'next-seo'
7
8 export default function Home() {
9   return (
10    <>
11      <NextSeo
12        title="Descomplique a sua agenda | Oducoelho Call"
13        description="Conecte seu calendário e permita que as pessoas marquem agendamentos
14          no seu tempo livre.."
15      />
16      <Container>
17        <Hero>
18          <Heading size="4xl">Agendamento descomplicado</Heading>
19          <Text size="xl">
20            Conecte seu calendário e permita que as pessoas marquem agendamentos
21            no seu tempo livre.
22          </Text>
23
24          <ClaimUsernameForm />
25        </Hero>
26
27        <Preview>
28          <Image
29            src={previewImage}
30            height={400}
31            quality={100}
32            priority
33            alt="Calendário simbolizando a aplicação em funcionamento"
34          />
35        </Preview>
36      </Container>
37    </>
38  )
39 }
40
```

Resultado da código na web:

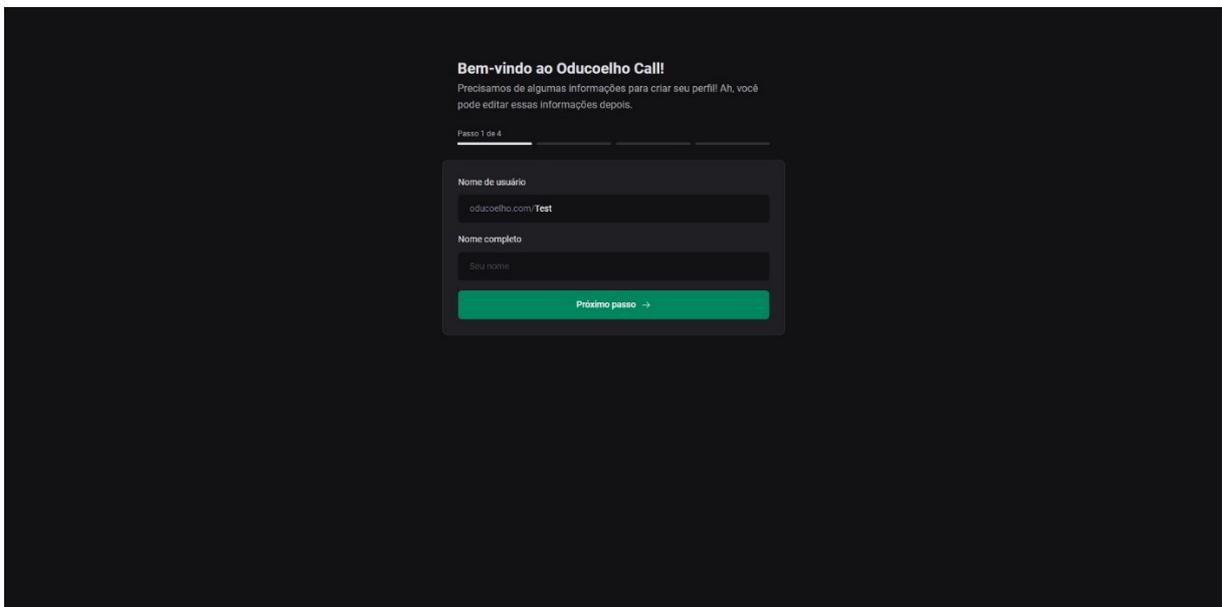
The screenshot shows a dark-themed web application. On the left, there is a main heading "Agendamento descomplicado" in white. Below it, a sub-heading reads "Conecte seu calendário e permita que as pessoas marquem agendamentos no seu tempo livre." There is a text input field containing "oducoelho.com/ seu-usuario" and a green "Reservar" button with a right-pointing arrow. Below the input field, a small text prompt says "Digite o nome do usuário desejado".

On the right side, there is a calendar for "Setembro 2022". The calendar grid shows days from 1 to 30. The current date, "terça-feira, 20 de setembro", is highlighted. To the right of the calendar is a vertical list of time slots: 9:00h, 10:00h, 11:00h, 12:00h, 13:00h, 14:00h, 15:00h, and 16:00h. Each time slot is represented by a dark grey rectangular button.

Para a página de cadastro de usuário foi utilizado o seguinte código:

```
1 import { zodResolver } from '@hookform/resolvers/zod'
2 import { Button, Heading, MultiStep, Text, TextInput } from '@ignite-ui/react'
3 import { AxiosError } from 'axios'
4 import { useRouter } from 'next/router'
5 import { ArrowRight } from 'phosphor-react'
6 import { useEffect } from 'react'
7 import { useForm } from 'react-hook-form'
8 import { z } from 'zod'
9 import { api } from '../../lib/axios'
10 import { Container, Form, FormError, Header } from './styles'
11 import { NextSeo } from 'next-seo'
12
13 const registerFormSchema = z.object({
14   username: z
15     .string()
16     .min(3, { message: 'O usuário precisa ter pelo menos 3 letras.' })
17     .TestRegex(
18       .regex(/^[a-z\-\-]+$/i, {
19         message: 'O usuário pode ter apenas letras e hifens.',
20       })
21     )
22     .transform((username) => username.toLowerCase()),
23   name: z
24     .string()
25     .min(3, { message: 'O nome precisa ter pelo menos 3 letras.' })
26 })
27
28 type RegisterFormData = z.infer<typeof registerFormSchema>
29
30 export default function Register() {
31   const {
32     register,
33     handleSubmit,
34     setValue,
35     formState: { errors, isSubmitting },
36   } = useForm<RegisterFormData>({
37     resolver: zodResolver(registerFormSchema),
38   })
39
40   const router = useRouter()
41
42   useEffect(() => {
43     if (router.query?.username) {
44       setValue('username', String(router.query.username))
45     }
46   }, [router.query?.username, setValue])
47
48   async function handleRegister(data: RegisterFormData) {
49     try {
50       await api.post('/users', {
51         name: data.name,
52         username: data.username,
53       })
54
55       await router.push('/register/connect-calendar')
56     } catch (err) {
57       if (err instanceof AxiosError && err?.response?.data?.message) {
58         alert(err.response.data.message)
59       }
60       return
61     }
62   }
63
64   console.log(err)
65
66   return (
67     <>
68       <NextSeo title="Crie uma conta | Oducoelho Call" />
69       <Container>
70         <Header>
71           <Heading as="strong">Bem-vindo ao Oducoelho Call!</Heading>
72           <Text>
73             Precisamos de algumas informações para criar seu perfil! Ah, você
74             pode editar essas informações depois.
75           </Text>
76           <MultiStep size={4} currentStep={1} />
77         </Header>
78
79         <Form as="form" onSubmit={handleSubmit(handleRegister)}>
80           <label>
81             <Text size="sm">Nome de usuário</Text>
82             <TextInput
83               prefix="oducoelho.com/"
84               placeholder="seu usuário"
85               {...register('username')}
86             />
87
88             {errors.username && (
89               <FormError size="sm">{errors.username.message}</FormError>
90             )}
91           </label>
92
93           <label>
94             <Text size="sm">Nome completo</Text>
95             <TextInput placeholder="Seu nome" {...register('name')} />
96
97             {errors.name && <FormError>{errors.name.message}</FormError>
98           </label>
99
100           <Button type="submit" disabled={isSubmitting}>
101             Próximo passo
102           </Button>
103         </Form>
104       </Container>
105     </>
106   )
107 }
```

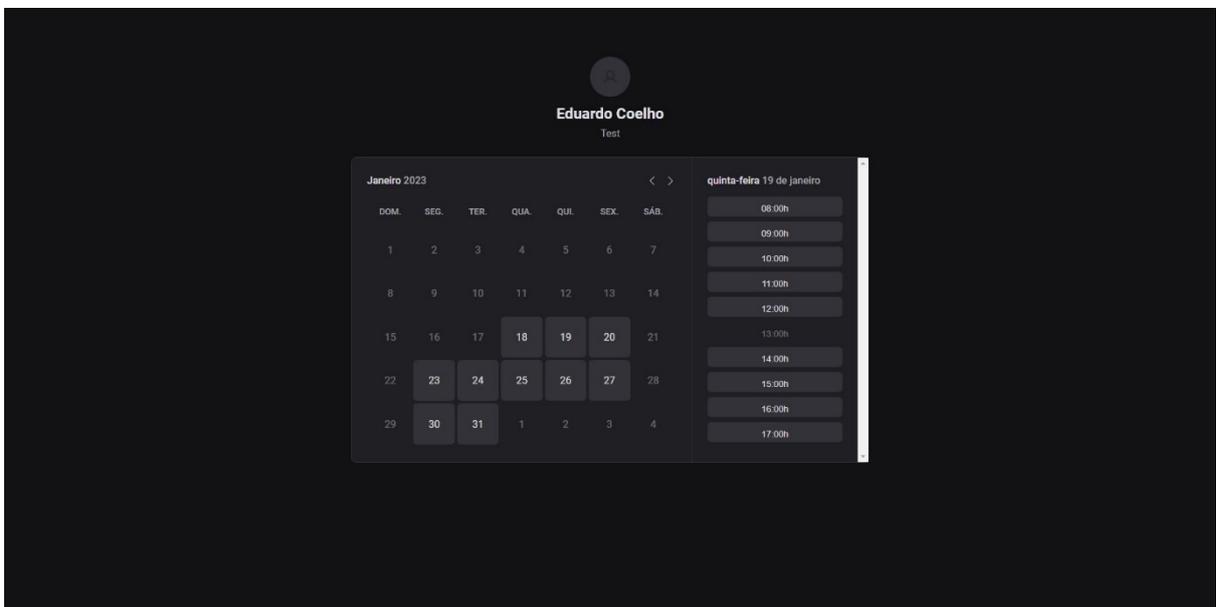
Resultado da código na web:



Para a página de agendamento foi utilizado o seguinte código:

```
1 import { Avatar, Heading, Text } from '@ignite-ui/react'
2 import { GetStaticPaths, GetStaticProps } from 'next'
3 import { NextSeo } from 'next-seo'
4 import { prisma } from '../../lib/prisma'
5 import { ScheduleForm } from './ScheduleForm'
6 import { Container, UserHeader } from './styles'
7
8 interface ScheduleProps {
9   user: {
10     name: string
11     bio: string
12     avatarUrl: string
13   }
14 }
15
16 export default function Schedule({ user }: ScheduleProps) {
17   return (
18     <NextSeo title='Agendar com ${user.name} | Oducoelho Call' />
19     <Container>
20       <UserHeader>
21         <Avatar src={user.avatarUrl} />
22         <Heading>{user.name}</Heading>
23         <Text>{user.bio}</Text>
24       </UserHeader>
25
26       <ScheduleForm />
27     </Container>
28   )
29 }
30
31 export const getStaticPaths: GetStaticPaths = async () => {
32   return {
33     paths: [],
34     fallback: 'blocking',
35   }
36 }
37
38 export const getStaticProps: GetStaticProps = async ({ params }) => {
39   const username = String(params?.username)
40
41   const user = await prisma.user.findUnique({
42     where: {
43       username,
44     },
45   })
46
47   if (!user) {
48     return {
49       notFound: true,
50     }
51   }
52
53   return {
54     props: {
55       user: {
56         name: user.name,
57         bio: user.bio,
58         avatarUrl: user.avatar_url,
59       },
60     },
61     revalidate: 60 * 60 * 24, // 1 day
62   }
63 }
64
65
66
```

Resultado da código na web:



### 3.3 SISTEMAS OPERACIONAIS

Ao utilizar os conceitos de Sistemas Operacionais, como gerenciamento de processos, gerenciamento de memória e escalonamento de recursos, o sistema operacional Linux auxilia na resolução do problema identificado, que é a necessidade de um ambiente estável, seguro e flexível para hospedar o website de comércio eletrônico. Essa tecnologia proporciona um ambiente confiável e eficiente para executar os processos do servidor web, gerenciar a alocação de recursos e garantir que o website esteja sempre disponível e responsivo para os usuários.

#### 3.3.1 COMPONENTES DE SISTEMAS OPERACIONAIS

Existem diferentes sistemas operacionais que poderiam ser considerados para aplicabilidade. Aqui estão alguns exemplos:

**Windows Server:** O Windows Server é outra opção comumente utilizada para hospedagem de websites. Ele oferece integração com tecnologias Microsoft, como ASP.NET e SQL Server, o que pode ser relevante se o website de comércio eletrônico for desenvolvido nesse ecossistema.

**Unix:** Unix é um sistema operacional precursor do Linux e ainda é amplamente utilizado em ambientes de servidores. Variantes do Unix, como FreeBSD e OpenBSD, podem ser consideradas para hospedagem de websites de comércio eletrônico, especialmente se houver necessidade de recursos específicos do Unix.

**macOS Server:** Embora menos comum em ambientes de servidores web, o macOS Server, baseado no sistema operacional macOS da Apple, pode ser uma opção a ser considerada se o projeto envolver integração com outros serviços ou aplicativos da Apple.

É importante levar em consideração os requisitos técnicos do projeto, as preferências da equipe de desenvolvimento e as necessidades específicas do website de agendamentos ao escolher o sistema operacional mais adequado. Cada um desses sistemas operacionais possui suas próprias características e recursos, portanto, é fundamental avaliar cuidadosamente suas vantagens e desvantagens em relação aos requisitos do projeto.

#### 3.3.2 GERENCIAMENTO E FUNCIONALIDADES DO SISTEMA OPERACIONAL

O sistema operacional Linux possui várias vantagens que o tornam uma escolha popular para hospedagem de servidores web. No entanto, é importante compará-lo com outros sistemas operacionais para uma avaliação completa das

opções disponíveis. Aqui estão algumas vantagens e comparações com outros sistemas operacionais:

### **Vantagens do Linux:**

**1. Estabilidade:** O Linux é conhecido por sua estabilidade e confiabilidade, oferecendo um ambiente robusto para hospedar aplicações web. Ele é projetado para executar por longos períodos sem necessidade de reinicialização, minimizando o tempo de inatividade.

**2. Segurança:** O Linux tem uma reputação sólida em termos de segurança. A comunidade de código aberto está constantemente revisando e aprimorando o sistema, o que resulta em atualizações regulares e correções rápidas para vulnerabilidades de segurança.

**3. Flexibilidade:** O Linux suporta uma ampla gama de linguagens de programação, bancos de dados e aplicativos, oferecendo flexibilidade para personalizar o ambiente de acordo com as necessidades específicas do projeto. Além disso, é altamente modular e pode ser facilmente adaptado e configurado conforme necessário.

### **Comparação com outros sistemas operacionais:**

#### **1. Windows Server:**

**Vantagens:** Integração com tecnologias Microsoft, como ASP.NET e SQL Server, que podem ser relevantes se o projeto utilizar essas tecnologias. Interface de usuário familiar para usuários do Windows.

**Desvantagens:** Menos comum em ambientes de servidores web. Pode exigir licenciamento adicional para algumas versões e recursos.

#### **2. Unix:**

**Vantagens:** Variantes do Unix, como FreeBSD e OpenBSD, têm uma longa história de estabilidade e segurança. São conhecidos por terem um bom desempenho em ambientes de servidor.

**Desvantagens:** Requer conhecimentos técnicos especializados para administração e configuração. Menos suporte de hardware em comparação com o Linux.

#### **3. macOS Server:**

**Vantagens:** Integração com outros dispositivos e serviços da Apple. Interface de usuário intuitiva para usuários do macOS.

**Desvantagens:** Menos comum em ambientes de servidor web. Pode ter menos suporte de hardware e opções de personalização em comparação com outros sistemas operacionais.

No geral, o Linux se destaca em termos de estabilidade, segurança e flexibilidade, sendo amplamente utilizado para hospedagem de servidores web.

### **3.3.3 GERENCIAMENTO DE HARDWARE PELO SISTEMA OPERACIONAL**

O protótipo criado é uma representação funcional do sistema de hospedagem da aplicação de agendamentos. Ele foi desenvolvido para avaliar o desempenho do ambiente de hospedagem, incluindo o servidor web, o banco de dados e a segurança do sistema.

No que diz respeito à avaliação do ambiente de hospedagem, foram considerados fatores como a velocidade de resposta do servidor, a disponibilidade do sistema, a capacidade de lidar com picos de tráfego e a escalabilidade para suportar o crescimento futuro. Essa avaliação foi realizada por meio de testes de carga, monitoramento do tempo de resposta e análise do desempenho geral do sistema.

Quanto à base de dados do sistema, foram consideradas diferentes opções: um banco de dados local e um banco de dados em nuvem. Cada opção possui implicações distintas em questões de segurança, atualização e fluxo de dados.

No caso de um banco de dados local, os dados são armazenados e processados no servidor local, o que pode oferecer maior controle sobre a segurança dos dados, pois eles não são acessíveis pela internet. No entanto, a administração e a manutenção do banco de dados local podem exigir mais esforço, além de limitar o acesso remoto aos dados.

Por outro lado, um banco de dados em nuvem permite armazenar e acessar os dados de forma remota, o que pode ser conveniente em termos de acesso e compartilhamento de dados. Além disso, serviços em nuvem geralmente fornecem recursos de segurança robustos, como criptografia de dados em repouso e em trânsito, backups automatizados e medidas de proteção contra falhas de hardware.

No entanto, ao utilizar um banco de dados em nuvem, é necessário considerar questões de segurança relacionadas à proteção dos dados contra acessos não autorizados. Também é importante estar ciente das políticas de privacidade e conformidade do provedor de serviços em nuvem.

Em termos de atualizações, um banco de dados local requer que as atualizações sejam realizadas manualmente, enquanto um banco de dados em nuvem geralmente é atualizado automaticamente pelo provedor de serviços.

Quanto ao fluxo de dados, um banco de dados local pode ter um desempenho mais rápido devido à latência reduzida, já que os dados estão sendo

acessados localmente. No entanto, um banco de dados em nuvem permite maior flexibilidade e escalabilidade, facilitando a expansão do sistema conforme necessário.

Em resumo, as possíveis vantagens de um banco de dados local incluem maior controle sobre a segurança dos dados e menor dependência de conexões de internet estáveis. Já as vantagens de um banco de dados em nuvem incluem conveniência de acesso remoto, segurança avançada fornecida pelo provedor de serviços e maior escalabilidade. No entanto, é importante avaliar as necessidades e requisitos específicos do projeto antes de decidir qual abordagem de banco de dados é a mais adequada, levando em consideração questões de segurança, atualização e fluxo de dados.

## **4 CONCLUSÃO**

Durante o processo de desenvolvimento, enfrentamos desafios técnicos que exigiram soluções criativas e uma abordagem iterativa. A colaboração entre a equipe de desenvolvimento e o cliente foi fundamental para entender as demandas e realizar ajustes conforme o necessário. O sistema de agendamentos, é um sistema confiável, seguro e de fácil manuseio para o seu usuário.

## REFERÊNCIAS

Para mais informações sobre o website de agendamentos, segue o repositório no Github: <https://github.com/oducoelho/oducoelho-call>