



**JEAN ROBERTO LOPES CRUZ**

**DESENVOLVIMENTO WEB DE LANDING  
PAGES**

**LAVRAS – MG**

**2023**



**JEAN ROBERTO LOPES CRUZ**

**DESENVOLVIMENTO WEB DE LANDING PAGES**

Relatório de estágio apresentado à Universidade Federal de Lavras, como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel.

Profa. DSc. Marluce Rodrigues Pereira  
Orientadora

**LAVRAS – MG**

**2023**

**JEAN ROBERTO LOPES CRUZ**

**DESENVOLVIMENTO WEB DE LANDING PAGES**

Relatório de estágio apresentado à Universidade Federal de Lavras, como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel.

APROVADO em 17 de fevereiro de 2023.

Bel. Lucas Bessa - Empresa ioasys

Prof. DSc. Rafael Serapilha Durelli - UFLA

 Documento assinado digitalmente  
MARLUCE RODRIGUES PEREIRA  
Data: 25/02/2023 18:44:52-0300  
Verifique em <https://verificador.iti.br>

Profª. DSc. Marluce Rodrigues Pereira - UFLA  
(Orientadora)

LAVRAS - MG  
2023

*Dedico esse trabalho a meus pais e irmãs que sempre foram meu alicerce, e também a minhas avós Dona Cidinha e Dona Mercedes, fontes de inspiração eternas.*



## **AGRADECIMENTOS**

Agradeço a Deus por me dar forças diariamente, sendo meu amparo e refúgio.

Gratidão aos meus familiares que me apoiaram quando foi necessário, principalmente meus pais Jane e Roberto pelo amor incondicional a mim depositado.

Aos amigos que estiveram presentes ao meu lado nessa jornada, excepcionalmente: Bruno Amaral, Guilherme Soares, Gustavo Reis, Leonardo Campelo, Matheus Amâncio e Victor Cabral.

A Comp Júnior e as amizades verdadeiras lá construídas, que me ensinaram a desenvolver pessoas, fazer acontecer, amar resultados e ir além.

Agradeço a ioasys, por acreditar em mim, por contribuir em meu crescimento profissional e pessoal, além de expandir meus horizontes.

Aos docentes e técnicos do Departamento de Ciência da Computação, em especial a professora Marluce, pela oportunidade de orientação e o compartilhamento de conhecimentos.

A Universidade Federal de Lavras, por proporcionar que meu sonho de cursar uma graduação fosse realizado.



*"O temor do Senhor é o princípio da Sabedoria, e o conhecimento do Santo é o entendimento."*

*(Provérbios, 9:10)*



## RESUMO

Este documento descreve as atividades realizadas durante estágio na empresa io-asy. O objetivo principal foi atuar como desenvolvedor web front-end em diversos projetos, todos com o propósito de desenvolver landing pages. O relatório visa apresentar os processos de desenvolvimento de software dentro da empresa e discorrer sobre as atividades realizadas que permitiram que o estagiário aplicasse os conhecimentos adquiridos no decorrer do curso de graduação. Além de explicitar sobre o aprendizado de ferramentas, bem como conhecer novas tecnologias, aplicar os conceitos da metodologia ágil Scrum e relatar as experiências obtidas em toda essa jornada.

**Palavras-chave:** Front-end. Landing page. Desenvolvimento web.



## **ABSTRACT**

This document describes the activities carried out during the internship at the company ioasys. The main objective was to act as a front-end web developer in several projects, all with the purpose of developing landing pages. The report aims to present the software development processes within the company and discuss the activities carried out that allowed the intern to apply the knowledge acquired during the under graduation course. In addition to explaining about learning tools, as well as getting to know new technologies, applying the concepts of the agile Scrum methodology and reporting on the experiences obtained throughout this journey.

**Keywords:** Front-end. Landing page. Web development.



## LISTA DE FIGURAS

Figura 2.1 – Processos e pessoas de Scrum . . . . .	20
Figura 2.2 – Kanban de um dos projetos usando a ferramenta TeamWork . . . . .	22
Figura 3.1 – Página <i>web</i> de autenticação criada para o projeto fictício . . . . .	33
Figura 3.2 – Ciclo de vida ideal de um projeto . . . . .	35
Figura 3.3 – <i>Mockups</i> em versões <i>desktop</i> e <i>mobile</i> para página de um dos projetos. . . . .	37
Figura 3.4 – Página <i>web</i> criada para um dos projetos . . . . .	39
Figura 3.5 – Seção “Sobre o evento” . . . . .	39
Figura 3.6 – Seção “Palestrantes” . . . . .	40
Figura 3.7 – Seção “Programação” . . . . .	41
Figura 3.8 – Seção “Contato” . . . . .	41



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	17
<b>2</b>	<b>METODOLOGIAS E FERRAMENTAS</b>	19
<b>2.1</b>	<b>Metodologia ágil Scrum</b>	19
<b>2.1.1</b>	<b>Product Backlog</b>	20
<b>2.1.2</b>	<b>Kanban</b>	21
<b>2.1.3</b>	<b>Sprint</b>	21
<b>2.1.4</b>	<b>Sprint Backlog</b>	22
<b>2.1.5</b>	<b>Sprint Planning</b>	22
<b>2.1.6</b>	<b>Daily Scrum</b>	23
<b>2.1.7</b>	<b>Sprint Review</b>	23
<b>2.1.8</b>	<b>Sprint Retrospective</b>	24
<b>2.1.9</b>	<b>Scrum Team</b>	24
<b>2.2</b>	<b>Ferramentas</b>	25
<b>2.2.1</b>	<b>IDE Visual Studio Code</b>	25
<b>2.2.2</b>	<b>Bitbucket</b>	25
<b>2.2.3</b>	<b>Teamwork</b>	25
<b>2.3</b>	<b>Tecnologias</b>	26
<b>2.3.1</b>	<b>Front-end</b>	26
<b>2.3.2</b>	<b>JavaScript</b>	26
<b>2.3.3</b>	<b>TypeScript</b>	26
<b>2.3.4</b>	<b>HTML</b>	28
<b>2.3.5</b>	<b>CSS</b>	28
<b>2.3.6</b>	<b>ReactJS</b>	29
<b>2.3.7</b>	<b>GatsbyJS</b>	29
<b>3</b>	<b>ATIVIDADES DESENVOLVIDAS</b>	31
<b>3.1</b>	<b>Processo seletivo</b>	31
<b>3.2</b>	<b>Mentoria</b>	32

<b>3.3</b>	<b>Projetos . . . . .</b>	<b>33</b>
<b>4</b>	<b>CONCLUSÃO . . . . .</b>	<b>43</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>47</b>

## 1 INTRODUÇÃO

O avanço da tecnologia da informação mudou a forma das transações de compra e venda no mundo todo. Para sobreviver no mercado, as organizações têm buscado cada vez mais as técnicas de *marketing* digital.

O *marketing* digital visa promover produtos ou marcas por meio de mídias digitais (CARVALHO, 2021). Existem alguns métodos online usados para guiar o tráfego dos clientes, como: otimização de máquinas de busca (o site aparece próximo ao topo da lista de busca retornada pela máquina de busca), propagandas pagas que aparecem do lado do resultado das máquinas de busca (*pay-per-click*), propagandas no site com *banners*, texto e mídia avançada, *landing pages*, sites em redes sociais, listas de e-mails, blogs, clipes de vídeos, entre outros.

*Landing pages* são páginas de um site estático que tem como foco principal a conversão dos visitantes em possíveis clientes (BALLARIN, 2020). Recurso chave nas campanhas de *marketing* digital, essas páginas de captura possuem menos elementos que uma página tradicional, focando apenas no produto da oferta central para atrair o possível cliente, no caso o visitante da página. Quando tal visitante efetua uma compra, um download ou preenche um formulário completo do site, há uma conversão de um possível interessado em cliente que gera valor para o negócio. Portanto, possuem grande importância nos sites e devem ser projetadas com foco na atração do cliente. Como exemplo, pode-se citar as *landing pages* das empresas Airbnb<sup>1</sup>, Slack<sup>2</sup> e Uber<sup>3</sup>.

Uma *landing page* pode fazer parte do site principal ou ser projetada especificamente para ser focada em um público em particular.

Este documento relata as atividades desenvolvidas durante o estágio de 900 horas desempenhado na empresa **ioasys**<sup>4</sup>, empresa brasileira de inovação tec-

---

<sup>1</sup> Airbnb: <<https://www.airbnb.com.br/host/homes>>

<sup>2</sup> Slack: <<https://slack.com/intl/pt-br/lp>>

<sup>3</sup> Uber: <<https://www.uber.com/br/pt-br/drive>>

<sup>4</sup> ioasys: <<https://ioasys.com.br/>>

nológica especializada na criação e desenvolvimento de aplicativos móveis e *web*. A atuação ocorreu como programador *front-end* em um time composto por programadores da mesma área, os quais integraram vários projetos de curta duração que tiveram como ponto em comum o desenvolvimento web de *landing pages*.

Na ioasys, para realização de projetos de desenvolvimento de *landing pages*, técnicas e práticas para gerir projetos foram determinadas por metodologia ágil Scrum (SUTHERLAND, 2014). Seus processos, organização de atividades, definição de funções a serem exercidas e interação entre os membros foram alguns dos itens utilizados, além de adaptações de ritos moldados conforme as necessidades do time.

Além disso, houve o uso de tecnologias utilizadas com o intuito de facilitar um desenvolvimento em conjunto por componentes reutilizáveis, versionamento de código, constantes revisões e manutenções.

É importante explicitar sobre conceitos de desenvolvimento essenciais para entendimento da estrutura de uma aplicação web, bem como *front-end*, *frameworks* e linguagens de programação.

O texto também visa discorrer sobre as trocas de experiências, tanto o conhecimento obtido durante as disciplinas no curso de Ciência da Computação e consequentemente aplicadas nas tarefas desempenhadas, quanto nas novas práticas proporcionadas pelo dia a dia na organização.

Por fim, demonstrar resultados obtidos por meio dos projetos finalizados e entregues com sucesso, mostrando características comuns entre eles.

Os demais capítulos deste presente documento estão organizados como segue. O Capítulo 2 apresenta metodologias utilizadas durante o estágio e também ferramentas presentes para o desenvolvimento dos websites. Já no Capítulo 3 visa mostrar as experiências vivenciadas no cotidiano da qual o estagiário fez parte. O Capítulo 4 apresenta comentários e observações finais acerca do curso de Ciência da Computação e o estágio.

## 2 METODOLOGIAS E FERRAMENTAS

### 2.1 Metodologia ágil Scrum

Para o desenvolvimento dos projetos foram utilizadas metodologia ágil e ferramentas para auxiliar no gerenciamento do mesmo e implementação. Para melhor entendimento, as seções a seguir abordarão cada um desses pontos.

As metodologias ágeis são abordagens para o desenvolvimento de produtos que estão alinhadas com os valores e princípios descritos no Manifesto Ágil para Desenvolvimento de *Software* (BECK et al., 2001), assinado em 2001, o qual reuniu líderes de diferentes empresas de tecnologia que trouxeram consigo abordagens distintas, porém compartilhando dos mesmos fundamentos para desenvolverem seus respectivos *softwares*.

Com o Manifesto, passou-se a valorizar os seguintes princípios para desenvolvimento ágil de *software*:

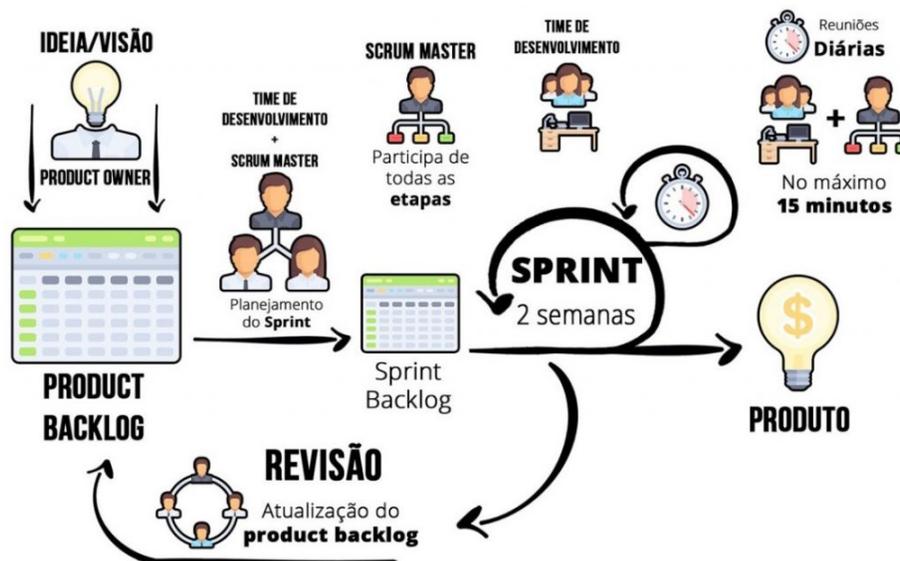
Indivíduos e interações mais que processos e ferramentas;  
*Software* em funcionamento mais que documentação abrangente;  
Colaboração com o cliente mais que negociação de contratos;  
Responder a mudanças mais que seguir um plano  
(BECK et al., 2001).

Scrum é uma das metodologias ágeis mais empregadas nas organizações. Scrum é fruto do trabalho de Ken Schwaber e Jeff Sutherland, que a criaram com base em sua experiência em projetos de *software* complexos nos anos 80 e 90. Eles compartilharam seus conhecimentos sobre Scrum por um artigo publicado em 1995 e em seguida, lançaram um livro intitulado "Scrum Development Process" em 2001 (SUTHERLAND, 2014).

Com o intuito de melhorar o gerenciamento de projetos de *software*, o Scrum baseia-se em uma colaboração contínua, adaptação e entrega incremental. Ele é composto por artefatos, sendo eles *Product Backlog* e *Sprint Backlog*, seus eventos ou ritos, também conhecidos como *Sprint Planning*, *Daily Scrum*, *Sprint Review*, *Sprint Retrospective* e *Sprint*. e por fim, seus três papéis fundamentais, os

quais compõem o *Scrum Team* que são *Scrum Master*, *Product Owner* e o time de desenvolvimento (SCRUM, 2023). Além desses, para o auxílio no gerenciamento de forma explícita das atividades, foi utilizado a metodologia como Kanban. A seguir é explicado detalhadamente cada um dos pontos citados.

Figura 2.1 – Processos e pessoas de Scrum



Fonte:

<<https://engenharia360.com/scrum-a-metodologia-utilizada-na-gestao-de-projetos/>>

### 2.1.1 Product Backlog

O *Product Backlog* é uma lista de todos os requisitos que precisam estar presentes no produto, onde tal lista é continuamente atualizada e modificada. Pode ser organizada conforme a importância e com seu devido detalhamento, ou pode ser que alguns itens ainda não estejam claros. Isso permite que o projeto seja flexível e adaptável às mudanças, característica dos projetos Scrum. A responsabilidade de gerenciar, definir e ordenar os itens do *Product Backlog* é do *Product Owner*. No contexto do projeto, o *Scrum Master* também cumpre o papel de auxiliar nas definições de prioridade.

### 2.1.2 Kanban

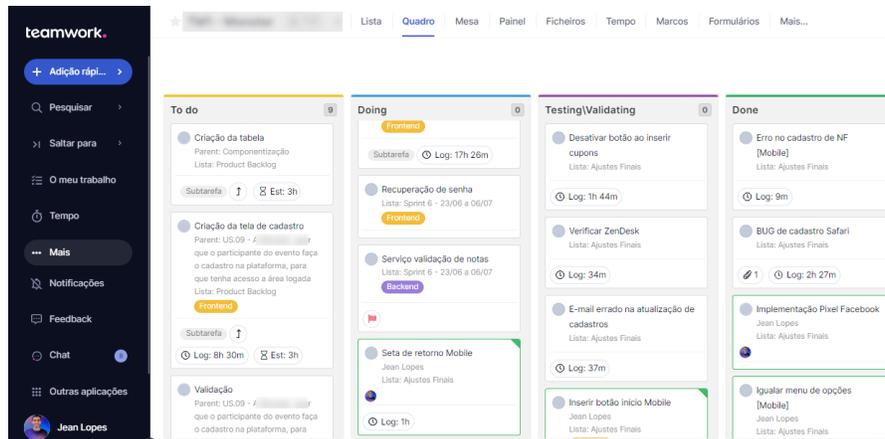
Kanban é uma abordagem de gerenciamento de fluxo de trabalho, que em sua origem foi utilizada para controlar a produção em fábricas, e posteriormente adaptada para gerenciar projetos e processos de negócios. Planeja tornar transparente o fluxo de trabalho, estabelecer limites para o trabalho em andamento e regular a velocidade de entrega para aumentar a eficiência e eficácia, permitindo que os membros da equipe vejam o estado de cada parte do trabalho a qualquer momento (ATLASSIAN, 2023b). A combinação de Kanban e Scrum para os projetos foi uma escolha inteligente, pois se trata de uma abordagem poderosa para gerenciamento dos mesmos, combinando as melhores características de ambas as metodologias.

Também altamente adaptável, o painel do Kanban do projeto é algo similar ao que se vê na Figura 2.2. As tarefas na coluna "*To Do*" são aquelas selecionadas a fazerem parte da *Sprint Backlog* e que ainda não estão sendo realizadas. Na coluna "*Doing*" aparecem as tarefas que já possuem atribuição para um integrante do time de desenvolvimento que está executando a mesma. Já na coluna de "*Testing/Validating*" surgem as tarefas executadas pelo time de desenvolvimento e estão aptas para que o *Product Owner* possa revisar se cumpre com sua finalidade proposta. Por fim, em "*Done*" ficam as tarefas que foram realizadas e revisadas com sucesso. Caso haja algum impedimento ou a tarefa não cumpra com o proposto, a mesma é priorizada a ser solucionada e, permanecendo insuficiente, ela retorna ao início do Kanban, possivelmente fazendo parte de uma nova *Sprint*.

### 2.1.3 Sprint

As metodologias ágeis de desenvolvimento de *software* são iterações curtas, no caso dos projetos da *ioasys* com duração de duas semanas, as quais o time de desenvolvimento trabalha planejando, implementando e entregando o trabalho.

Figura 2.2 – Kanban de um dos projetos usando a ferramenta TeamWork



Tal iteração é chamada de *Sprint*, sendo ela o principal evento de Scrum, pois são nessas iterações que as tarefas são realizadas e o produto é desenvolvido.

Durante a *Sprint*, o time se compromete a entregar um incremento funcional do produto, e ao final de cada iteração é realizado uma revisão para avaliar o progresso e planejar a próxima *Sprint*.

#### 2.1.4 Sprint Backlog

*Sprint Backlog* é uma lista de tarefas que a equipe de desenvolvimento se compromete a realizar durante uma *Sprint*, criado durante a *Sprint Planning*. Passa por revisão e atualização diariamente durante a *Sprint*, por vezes nas reuniões *Daily* é compartilhado em tela o Kanban da respectiva *Sprint*, sendo usado para acompanhar o progresso e identificar problemas ou impedimentos.

#### 2.1.5 Sprint Planning

A reunião de planejamento da *Sprint* é usada para organizar o trabalho a ser realizado na próxima iteração. Durante essa reunião, as funcionalidades do *Product Backlog* são escolhidas e priorizadas, a fim de que todo o time de de-

envolvimento tenha conhecimento das funcionalidades a serem implementadas e estabeleça um plano para concluí-las.

Durante a reunião de *Sprint Planning*, o *Product Owner* é quem conduz a mesma e aponta no *Product Backlog* as funcionalidades a serem desenvolvidas. Na ioasys, além de funcionar dessa forma, a equipe participa ativamente da reunião, buscando sanar suas dúvidas e dividir as funcionalidades das histórias em tarefas técnicas. Por fim, as tarefas selecionadas compõem o *Sprint Backlog*.

### **2.1.6 Daily Scrum**

O objetivo da *Daily Scrum* é uma reunião diária que visa inspecionar o progresso da *Sprint* e conseqüentemente atualizar o *Sprint Backlog* conforme necessário, dando andamento no projeto e viabilizando o alinhamento entre os integrantes do *Scrum Team* para a continuidade na realização das tarefas ou atribuição para novas.

Para manter a agilidade, ela é realizada no mesmo horário nos dias úteis da *Sprint* e no mesmo local de sala virtual. Além disso, o rito não deve durar mais que 15 minutos, já que seu intuito é apenas repassar individualmente respectiva evolução, identificar possíveis impedimentos e agilizar na tomada de decisão.

Em situações como dos projetos realizados, haviam outras reuniões mais detalhadas sobre refinamentos de tarefas, replanejamento da *Sprint* e, inclusive, resolução de impedimentos levantados no decorrer da *Sprint*. Tais reuniões eram geralmente lideradas pelo *Product Owner* e/ou *Scrum Master* acompanhadas das demais pessoas envolvidas na atividade.

### **2.1.7 Sprint Review**

A *Review* tem como princípio reunir a equipe para apresentar e avaliar o progresso alcançado ao longo da *Sprint*, a fim de determinar se o objetivo foi alcançado. Na ioasys, tanto a condução da reunião quanto a determinação se os

respectivos objetivos foram atingidos é papel do *Product Owner*. Por vezes, também pode ser uma chance para abrir e receber comentários de clientes e usuários sobre tal evolução. Assim, é ideal que todos os envolvidos na entrega e validação do produto estejam presentes.

Por fim, o *Product Backlog* é reajustado conforme os resultados obtidos e é provável que já se definam os próximos passos da *Sprint* subsequente.

### **2.1.8 Sprint Retrospective**

O objetivo da *Retrospective* é discutir em equipe maneiras de aumentar a qualidade e eficácia. O *Scrum Team* deve inspecionar como foi a última *Sprint* em relação aos indivíduos, interações, processos, ferramentas e sua definição de feito. As suposições que os desencaminharam deverão ser identificadas e suas origens exploradas. O Time deverá colocar em pauta o que deu certo durante a *Sprint*, quais problemas foram encontrados, como esses problemas foram (ou não) resolvidos e o que podem fazer para não voltarem a acontecer. O Time deverá identificar as mudanças mais úteis para melhorar sua eficácia.

### **2.1.9 Scrum Team**

O *Scrum Master* faz a facilitação do *Scrum* para equipes maiores garantindo que a estrutura seja seguida. É uma pessoa com o compromisso de exercer os valores e práticas do *Scrum*, mas também deve manter a flexibilidade e a abertura para oportunidades de melhorar o fluxo de trabalho da equipe.

O *Product Owner* é responsável por maximizar o valor do produto resultante do trabalho do *Scrum Team*. Tais responsabilidades são: desenvolver e comunicar explicitamente a meta do produto; criar e comunicar claramente os itens do *Product Backlog*; solicitação de itens do *Product Backlog*; e, garantir que o *Product Backlog* seja transparente, visível e compreendido.

Os desenvolvedores são as pessoas do *Scrum Team* que estão comprometidas em criar qualquer aspecto de um incremento utilizável a cada *Sprint*. São sempre responsáveis por: criar um plano para o *Sprint Backlog*; incutir qualidade ao aderir a uma definição de pronto; adaptar seu plano a cada dia em direção ao objetivo do *Sprint*; e, responsabilizando-se mutuamente como profissionais.

## **2.2 Ferramentas**

### **2.2.1 IDE Visual Studio Code**

Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Inclui suporte para depuração, controle de versão integrado, ferramentas de desenvolvimento de interface do usuário e integração com outras ferramentas de desenvolvimento comuns (MICROSOFT, 2023b).

### **2.2.2 Bitbucket**

Bitbucket é um sistema de gerenciamento de código-fonte e ferramenta de colaboração para desenvolvimento de *software* (ATLASSIAN, 2023a). Ele fornece recursos como armazenamento de código-fonte, controle de versão, gerenciamento de tarefas e integração com outras ferramentas de desenvolvimento. Ele é especialmente popular para projetos de código aberto e equipes de desenvolvimento pequenas a médias.

### **2.2.3 Teamwork**

Teamwork é uma plataforma de gerenciamento de projetos e colaboração para equipes. Ele fornece recursos como gerenciamento de tarefas, controle de tempo, gerenciamento de documentos, gerenciamento de recursos e comunicação

em equipe. Ele é projetado para ajudar as equipes a gerenciar e completar projetos de maneira eficiente e colaborativa (TEAMWORK, 2022).

## 2.3 Tecnologias

### 2.3.1 Front-end

*Front-end* é a parte do desenvolvimento *web* responsável pela construção e design da camada visual do site ou aplicativo, pela interface com o usuário (UI) (SINGHAL, 2023). A criação de sites inclui o uso de HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) e JavaScript<sup>1</sup> para projetar e construir a interface do usuário, bem como a implementação de recursos interativos e dinâmicos. O papel principal do desenvolvimento front-end é promover boa usabilidade e garantir que o software desenvolvido realmente funcione conforme o esperado.

### 2.3.2 JavaScript

*JavaScript* é uma linguagem de programação interpretada, principalmente utilizada para adicionar funcionalidades interativas e dinâmicas em páginas *web*. É usada para criar e controlar elementos HTML e CSS, bem como realizar operações de lógica e manipulação de dados. É uma das principais linguagens de programação para desenvolvimento *front-end* (MOZILLA, 2023b).

### 2.3.3 TypeScript

*TypeScript* é uma linguagem de programação fortemente tipada construída sobre JavaScript, desenvolvida e mantida pela Microsoft. É uma super-set de *JavaScript*, o que significa que todo o código JavaScript é código válido *TypeScript*. Ela adiciona recursos como tipagem estática, classes e interfaces, para melhorar

---

<sup>1</sup> <https://www.w3schools.com/css/default.asp>

a escalabilidade e manutenibilidade dos aplicativos desenvolvidos (MICROSOFT, 2023a).

Segundo (BOGNER; MERKEL, 2022), após análise de mais de 5000 repositórios no GitHub, os resultados mostraram que, em geral, as aplicações escritas em TypeScript tiveram melhor qualidade de *software* do que as escritas em JavaScript. As métricas de qualidade, como a manutenibilidade e a legibilidade do código, foram significativamente melhores para as aplicações escritas em TypeScript. Além disso, os autores também descobriram que a adoção de *TypeScript* em projetos de *software* está aumentando continuamente e recomendam o uso da mesma.

O exemplo de código a seguir é a implementação de uma função em *JavaScript* para o cálculo da média aritmética entre dois valores informados pelo usuário. Como *JavaScript* não sabe qual é o tipo de uma variável até que seja realmente instanciada em tempo de execução, a chamada da função **media** passando os valores 10 e "30", somente apresentará o erro que "30" não é um inteiro somente durante a exibição do resultado no navegador.

```
function media(valor1, valor2){  
  return (valor1 + valor2)/2  
}  
media(10, "30")
```

*TypeScript* adiciona suporte de tipo para JavaScript e captura erros de tipo durante a compilação. Se passarmos esse código para Typescript, a mensagem de erro é exibida dizendo que os dois valores passados como parâmetro na função deveriam ser números e o segundo parâmetro passado foi uma string.

```
function media(valor1:number, valor2:number){  
  return (valor1 + valor2)/2  
}  
media(10, "30")
```

### 2.3.4 HTML

HTML (*Hypertext Markup Language*) é uma linguagem de marcação utilizada para estruturar o conteúdo de páginas *web*. É usada para criar elementos como títulos, parágrafos, links, imagens e tabelas, e também permite adicionar atributos a esses elementos para definir sua aparência e comportamento (MOZILLA, 2023c).

A fim de facilitar o desenvolvimento, melhorar a classificação em páginas de buscadores e aumentar a acessibilidade, foi feito o uso do conceito de HTML semântico, usando cada etiqueta de acordo com sua real finalidade.

### 2.3.5 CSS

CSS (*Cascading Style Sheets*) é uma linguagem de estilo usada para descrever a apresentação de documentos escritos em HTML ou XML (MOZILLA, 2023a). É usada para definir como os elementos HTML devem ser exibidos na tela, incluindo cor, layout, tamanho e posição. CSS permite separar a apresentação da estrutura do conteúdo, tornando o desenvolvimento de sites mais fácil de manter e escalar, e conseqüentemente resultando num visual mais atrativo.

Dos principais recursos de CSS, destacam-se um dos mais utilizados, importante para experiência do usuário e que esteve presente nos projetos da qual o estagiário participou, chama-se *media query*. Ele permite fazer com que a página encaixe seu *layout* a tamanhos de tela distintos, realizando a definição e configuração de estilo de acordo com cada dispositivo que especificar. Tal conceito de distribuição de elementos e conteúdos baseado no tamanho da tela é conhecido como responsividade.

Com isso, foi explorado o método e a boa prática de *Mobile first*<sup>2</sup> em todos os projetos. Vale a pena aplicar, pois em 2022 o número de usuários globais

---

<sup>2</sup> *Mobile first*: método que faz a criação de projetos *web* e sites dando início a implementação para telas de dispositivos móveis, posteriormente fazer adaptações para o *desktop* e outras plataformas.

da internet foi de 4,5 bilhões, sendo que mais de dois terços (67,1%) da população mundial agora usa um telefone celular (KEMP, 2023). Além de tudo, de 2017 a 2022 a relação entre o mercado de *smartphones* e *desktops*, desde quando os dispositivos móveis são mais comprados que os dispositivos de mesa, a diferença nunca foi tão grande: 59,66% e 40,34%, respectivamente (STATCOUNTER, 2023).

### 2.3.6 ReactJS

ReactJS é uma biblioteca *JavaScript* desenvolvida e mantida pelo Facebook (Meta). Ela é usada para construir interfaces de usuário complexas e dinâmicas, permitindo que os desenvolvedores criem componentes reutilizáveis e gerenciem facilmente o estado e a renderização dos componentes (META, 2023). É uma das principais escolhas para desenvolvimento de aplicativos *web* e *mobile* modernos, que em 2022 foi o mais popular entre as escolhas de *frameworks* para *front-end* (STACKOVERFLOW, 2023).

Segundo (AGGARWAL, 2018), o ReactJS tem poder e recursos para atender aos requisitos das tendências atuais no que se refere a facilidade, eficiência e maior acessibilidade, bem como já impacta a maneira como os aplicativos são escritos para a *web*.

### 2.3.7 GatsbyJS

*GatsbyJS* é uma biblioteca de código aberto baseada em ReactJS usada para criar sites estáticos rápidos e escaláveis. É uma plataforma de criação de sites moderna que combina a potência do ReactJS com a velocidade e eficiência de sites estáticos, criando toda uma nova rede de possibilidades (MATHEWS, 2023).

Tal biblioteca foi escolhida, pois é de fácil implementação para projetos em que seja necessário utilizar conteúdos de idiomas diferentes, fazendo com que o site atinja pessoas de diversas nacionalidades, o que era comum entre os clientes da empresa que requisitou o projeto.

Importante ressaltar que no contexto da época em que os projetos foram executados, ou seja, no ano de 2021, o NextJS<sup>3</sup> não era um framework tão popular como nos dias de hoje, principalmente no que diz respeito a construção de sites estáticos. No entanto, Gastby mostrou-se satisfatório em seus recursos, desempenho, documentação, ecossistema de *plugins* e a comunidade de desenvolvedores que o suportam.

---

<sup>3</sup> NextJS é um framework criado em cima do ReactJS que usa a renderização do lado do servidor

### 3 ATIVIDADES DESENVOLVIDAS

#### 3.1 Processo seletivo

O processo seletivo teve como passo inicial a inscrição *online*, que contou com milhares de inscritos, dos quais foram selecionados uma centena de candidatos. Após entrevistas, cerca de 60 candidatos foram designados para avançarem a fase presencial.

O *Camp*, como o processo seletivo foi nomeado, reuniu mais de 60 horas de conteúdo percorridas durante um mês, tendo como local das atividades a sede da empresa em Belo Horizonte/MG. Foram abordadas temáticas corriqueiras da ioasys como *design thinking*<sup>1</sup>, metodologias ágeis, arquitetura de sistemas, *front-end*, *back-end*, desenvolvimento *mobile*, além de receber dicas de como montar um *Pitch*<sup>2</sup>.

Na semana inicial, foram realizadas interações entre os candidatos com dinâmicas simulando metodologias e tecnologias utilizadas em projetos. Na sequência, os candidatos foram divididos em equipes, sendo exposto a um desafio que duraria todo o processo. Cada equipe deveria apresentar sua solução digital para um tema sorteado e, no fim, apresentar sua ideia para uma banca avaliadora de especialistas. Cada membro da equipe havia escolhido ainda na fase de entrevistas uma área de atuação, e no meu caso foi escolhida a opção *front-end*.

O desafio procedeu durante as semanas através de reuniões presenciais na empresa. Paralelamente, cada dia da semana um especialista da organização realizava um treinamento sobre determinada área de atuação. Por exemplo, as terças-feiras o tema era *front-end* e durante 6 horas do dia o assunto era aprofundado.

Acompanhando as equipes, havia um mentor que também compunha o time ioasys e auxiliava com dicas sobre o negócio. No caso de dúvidas técnicas,

---

<sup>1</sup> *Design Thinking*: metodologia de desenvolvimento de produtos e serviços focados nas necessidades, desejos e limitações dos usuários.

<sup>2</sup> *Pitch*: curta apresentação para vender uma ideia, projeto ou negócio, no intuito de despertar o interesse do telespectador, muitas vezes de investimento.

encaminhava para contato de especialistas e, além disso, cumpria o papel de avaliar a evolução de cada integrante da equipe.

Ao final da mentoria, cada equipe apresentou seu projeto desenvolvido e o *Pitch* mostrando seu valor de negócio. A equipe da qual o candidato fez parte não foi selecionada como vencedora, porém houve o convite para compor o time da ioasys, agora oficialmente como estagiário da empresa.

### 3.2 Mentoria

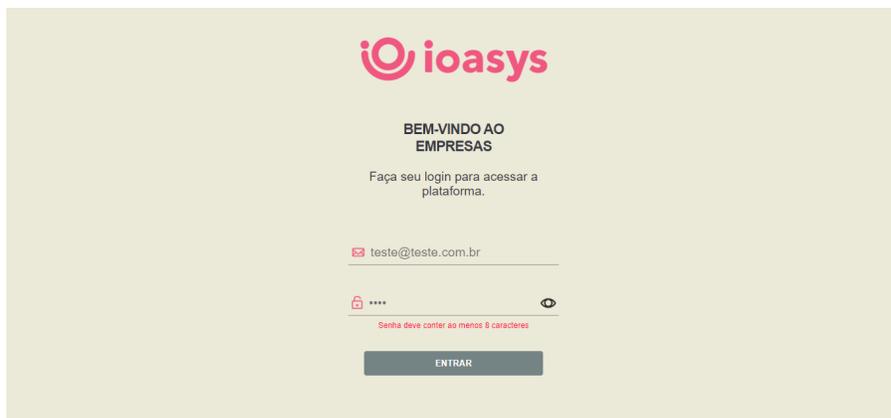
Nesse momento de entrada como estagiário, o objetivo principal era melhorar a proficiência em *front-end*. O desafio consistia na implementação de um site capaz de realizar autenticação e consultas numa *API*<sup>3</sup>, previamente disponibilizada pela equipe de especialistas de tecnologia da ioasys. Conjuntamente, a equipe de design transmitia quais eram os desafios quanto às interfaces web a serem desenvolvidas. Foi disponibilizada uma documentação detalhada sobre o *layout* com fluxo das páginas, cores, tamanhos de fontes e elementos, responsividade a serem seguidas, entre outros.

A Figura 3.1 apresenta uma página web de autenticação criada para o projeto fictício.

Uma das principais restrições do desafio passara pela utilização das tecnologias comuns da empresa, como ReactJS e realizar o envio dos códigos gerados para o repositório da empresa, no Bitbucket. Além disso, todo o código desenvolvido deveria estar nos padrões de qualidade da empresa. Dessa forma, todos os *Pull Requests*<sup>4</sup> que foram criados para incrementar o projeto, passara pela aprovação do mentor e também de outro desenvolvedor *front-end*. É válido ressaltar que o desafio não era limitado apenas ao desenvolvimento do projeto, e também em

<sup>3</sup> *API: Application Programming Interface*, em tradução livre: Interface de Programação de Aplicações. Significado: conjunto de definições e protocolos que permitem que serviços de tecnologia comuniquem entre si pela internet.

<sup>4</sup> *Pull request* é um mecanismo do *git* onde um desenvolvedor gera uma notificação que indica a conclusão do desenvolvimento de uma *feature* qualquer.

Figura 3.1 – Página *web* de autenticação criada para o projeto fictício

Fonte: Próprio do autor.

realizar as revisões dos códigos dos demais projetos, esses de clientes reais, desde que julgasse suficientemente capaz de analisar tal trecho de código. Havia um canal de comunicação exclusivo da empresa para comunicação sobre *Pull Requests* gerados nos projetos ativos.

Com isso, o repositório da empresa era utilizado recorrentemente, foi possível entender sobre padrões de código, aprofundar mais sobre *front-end* e realizar feedbacks constantes de melhoria e evolução técnica. Tal mentoria continha acompanhamento diário *online* de um mentor técnico, ao qual era desenvolvedor *front-end* sênior. A mentoria durou cerca de dois meses, finalizando com a entrega do desafio com sucesso.

### 3.3 Projetos

Após estar adaptado ao ambiente ioasys, houve alocação a um time de *front-end* para desenvolvimento de um projeto com clientes reais. O primeiro projeto foi de uma empresa de *streaming* de eventos *online*, especialista em conectar marcas aos seus clientes e funcionários. O desafio consistia em criar o site ins-

titucional dessa empresa, desenvolvendo, portanto, a primeira *landing page* do estagiário.

O *Scrum Team* (Seção 2.1.9) era composto por um *Scrum Master*, um *Product Owner*, três desenvolvedores *front-end* e uma *designer* de interfaces. A metodologia ágil fazia parte do nosso dia a dia, implicando em encontros diários da equipe respeitando os ritos do Scrum.

Sobre a utilização de tecnologias, eram presentes: ReactJS, *Styled Components*<sup>5</sup>, GatsbyJS e Git, usando as linguagens JavaScript e TypeScript. Mesmo que o conhecimento de muitas dessas tecnologias tivesse pouco tempo de contato ou nunca houvera contato antes, foi necessário que houvesse habilidades além de desenvolvimento técnico para acompanhar o ritmo da empresa. Lidar com um cliente importante no cenário brasileiro, que implica no acesso de milhares ou milhões de pessoas em sua plataforma foi de grande responsabilidade.

Entre o desenvolvimento, ajustes pontuais e entregas realizadas, foram dois meses até todas as telas ficarem totalmente conforme o cliente desejara e, finalmente, dar-se como finalizado o projeto.

Na sequência, ninguém da equipe foi alocado diretamente para um projeto, a espera ainda de detalhes burocráticos para essa nova inserção. Com isso, houve a proposta para estudar e aprofundar nas tecnologias as quais ainda não havia desenvoltura, como o próprio ReactJS e Typescript.

Depois de um mês, os estagiários foram alocados para o projeto de um evento *online*, que também passara pela entrega de uma *landing page* divulgando tal evento. Essa rotina de desenvolvimento de websites para eventos se repetiu por mais três vezes até finalizar o contrato com a organização como estagiário.

Na Figura 3.2, segue o fluxo ideal dos projetos nos quais o estagiário participou, considerando que em cor verde as etapas onde o próprio atuou diretamente.

---

<sup>5</sup> *Styled Components* é uma biblioteca que permite escrever códigos CSS dentro de *JavaScript*

Ainda sobre os projetos, os integrantes do *Scrum Team* eram os agentes que atuaram em cada uma das etapas. Complementar a esses, o cliente que foi, obviamente, quem demonstrou interesse e contratou para realizar a *landing page*, e é quem também arrematava a entrega do respectivo projeto no fim.

Figura 3.2 – Ciclo de vida ideal de um projeto



Fonte: Próprio do autor.

O fluxo inicia-se no passo 1 com *Scrum Master* e *Product Owner* realizando o levantamento dos requisitos do projeto. Na sequência, o designer alocado desenhava as telas que iriam compor o *layout* do site.

A Figura 3.3 mostra um exemplo de *mockup*<sup>6</sup> de como ficou a versão final do design de um dos projetos, o qual era de um evento promocional e que deveria ser desenvolvido para dispositivos na versão *desktop* e *mobile*.

Após aprovação do cliente, os requisitos eram colocados como tarefas e apresentados aos desenvolvedores, sendo esse o passo 3 na figura 3.2. Entre os passos de 3 a 6, eram os momentos em que os ritos de Scrum eram colocados em prática.

Depois das tarefas já estarem selecionadas e priorizadas, no passo 4 os desenvolvedores realizavam a implementação, que consistia em nada mais do que reproduzir as telas desenhadas, porém em forma de código utilizando HTML, CSS e JavaScript ou TypeScript. Já o passo 5 era colocado em prática após o envio das informações repassadas pelo cliente, compondo os elementos textuais e imagens do site.

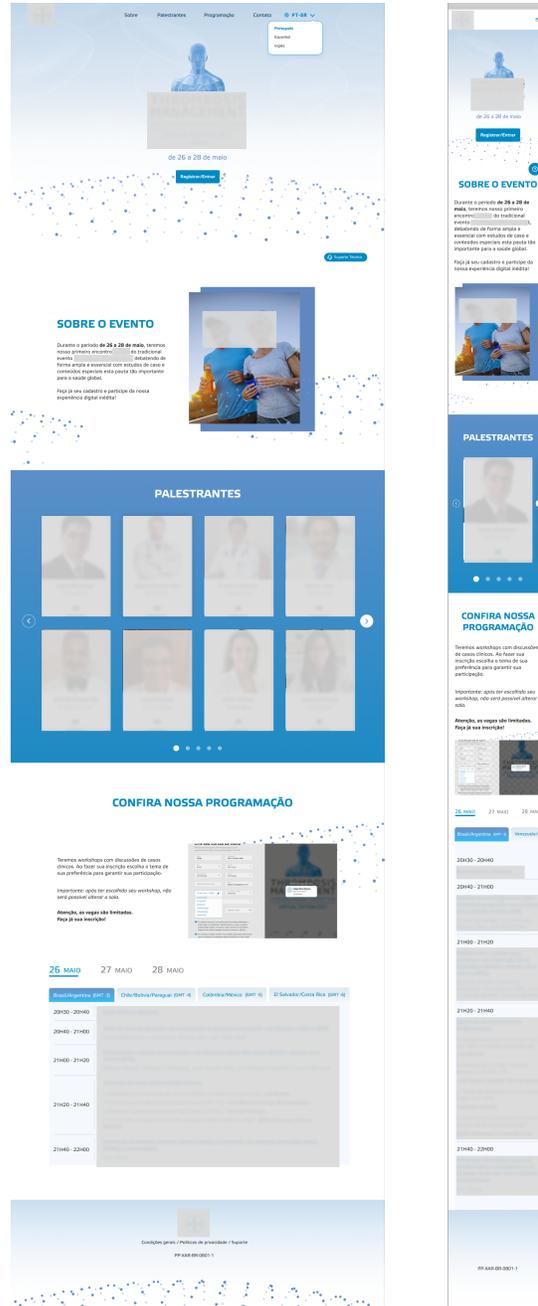
Após a conclusão da atividade de desenvolvimento e a mesma passar por revisão de código, a atividade seguia para o passo 6, a qual era testada e validada pelo *Product Owner* se estava dentro do esperado e se não possuía algum *bug*. Alguns bugs comuns eram a exibição de imagem desproporcional no tamanho na página, texto com erro ortográfico ou responsividade não funcionando para algum modelo de tela de dispositivo. Caso algum erro como os citados ou outro ocorresse, a tarefa voltava para a listagem de atividades para serem refeitas.

Por fim, com as tarefas aprovadas, o ambiente de produção era preparado para receber o código testado e o *Product Owner* liderava a reunião de entrega, realizando a exibição ao cliente e, posteriormente, o site ficava a disposição do

---

<sup>6</sup> *Mockup*: modelo simulado ou protótipo de uma interface de usuário, como um site ou aplicativo, que apresenta o aspecto final sem ter funcionalidade real.

Figura 3.3 – Mockups em versões *desktop* e *mobile* para página de um dos projetos.



Fonte: Equipe ioasys.

público para realizar acesso. Assim encerra-se o fluxo de ciclo de vida de um projeto ideal na ioasys, conforme mostrado na Figura 3.2.

Portanto, a *landing page* exercia função crucial de atrair o público interessado no assunto e incentivando os visitantes a se inscreverem no evento, gerando conhecimento, entretenimento, *networking*, expansão e divulgação de produtos.

Cada um dos projetos seguiu a sua aparência de acordo com seu respectivo objetivo. Em comum, os projetos tinham seções com informações sobre o evento, a lista de palestrantes, projetos ou produtos, a programação do evento e contato.

Nas Figuras 3.4, 3.5, 3.6, 3.7 e 3.8, seguem os exemplos de telas construídas no *mockup* validado pelo cliente em um dos projetos, o qual o estagiário participou e cujo intuito era apresentar informações sobre evento *online*, a fim de converter novos visitantes em participantes.

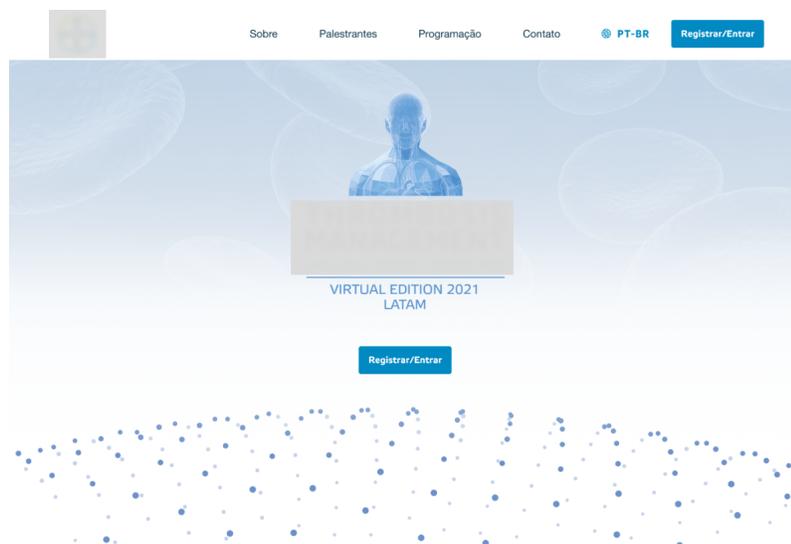
A primeira seção ou "seção de boas-vindas" (Figura 3.4) é a responsável por cativar a atenção do usuário e mantê-lo engajado, apresentando diretamente com elementos visuais impactantes qual o propósito da página, a fim de que o interessado realize a ação desejada, no caso, registrar-se no evento. Além disso, no cabeçalho possui um menu com as seções para facilitar a navegação e alteração de idioma.

A seção "Sobre o evento" (Seção 3.5) contém informações complementares aquelas que já foram introduzidas pela primeira seção. Visa aprofundar sobre o tema, como seus benefícios, depoimentos de quem já participou de outras edições. Dessa forma, gerando um interesse ainda maior por parte do usuário e contribuindo para tomar sua decisão de realizar a inscrição.

A seção "Palestrantes" (Figura 3.6) mostra os palestrantes com uma pequena biografia e experiência relevante para o evento, com o intuito do público se identificar com o tema abordado por eles.

A seção "Programação" (Figura 3.7) detalha o cronograma do evento, apresentando os dias e horários de cada uma das palestras e *workshops*. É útil

Figura 3.4 – Página web criada para um dos projetos



Fonte: Equipe ioasys.

Figura 3.5 – Seção “Sobre o evento”

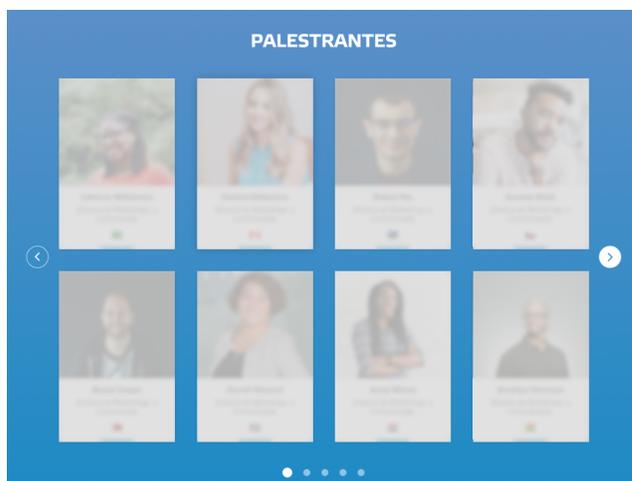


Fonte: Equipe ioasys.

a inclusão de uma breve explicação de como realizar a inscrição, um resumo do conteúdo do que será abordado e um informativo com a quantidade de vagas disponíveis, se for o caso.

A seção "Contato" (Figura 3.8) ou seção de rodapé permite que o usuário se comunique com os organizadores do evento de forma fácil e acessível, geralmente por meio de redes sociais, telefone ou e-mail. Além disso, pode conter os termos de uso, políticas de privacidade e suporte técnico.

Figura 3.6 – Seção “Palestrantes”



Fonte: Equipe ioasys.

A empresa ioasys desenvolve também outros tipos de sites, outras formas de tecnologia, inovação e design, não se restringindo em *landing pages* e sim sempre convergir no que diz respeito a cocriar experiências significativas para pessoas. Contudo, o *layout* de uma página como essa pode variar dependendo da estratégia de *marketing* e da natureza do produto ou serviço oferecido.

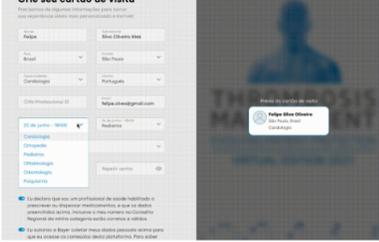
Figura 3.7 – Seção “Programação”

## CONFIRA NOSSA PROGRAMAÇÃO

Teremos workshops com discussões de casos clínicos. Ao fazer sua inscrição escolha o tema de sua preferência para garantir sua participação.

*Importante: após ter escolhido seu workshop, não será possível alterar a sala.*

**Atenção as vagas são limitadas. Faça já sua inscrição!**



26 MAIO    27 MAIO    28 MAIO

Brasil/Argentina (GMT -3)

Colômbia/México (GMT -5)

Fuso 03

20H30 - 20H40	
20H40 - 21H00	
21H00 - 21H20	
21H20 - 21H40	
21H40 - 22H00	

Fonte: Equipe ioasys.

Figura 3.8 – Seção “Contato”



Siga-nos

Condições gerais / Políticas de privacidade / Imprint / Chat e canais de atendimento

Fonte: Equipe ioasys.



## 4 CONCLUSÃO

Durante o período de sete meses de estágio, foi possível aplicar e fortalecer os princípios teóricos aprendidos nas diferentes disciplinas oferecidas pelo curso de Ciência da Computação da Universidade Federal de Lavras. Dentre as disciplinas cursadas, é válido ressaltar as principais contribuintes e seus conceitos, sendo elas:

- Engenharia de Software: ciclos de vida de um software, seus requisitos e sua qualidade, planejamento e gerência de projetos;
- Interação Humano-Computador: usabilidade, acessibilidade e experiência do usuário em interfaces;
- Linguagens de Programação: flexibilidade, confiabilidade e legibilidade para boas práticas;
- Estrutura de Dados: embasamento na escolha pela melhor estrutura e o manuseio da mesma conforme a situação apresentada.

Somado a isso, as experiências providas em sala de aula através de embasamentos teóricos de numerosas disciplinas se mostraram úteis, elevando a capacidade de resolução de problemas independente da complexidade com os desafios técnicos colocados a prova.

Na primeira etapa do estágio, a entrega do projeto fictício inicial foi um desafio tremendo no que diz respeito a autonomia e proatividade que exigiram adaptabilidade por parte do estagiário, considerando por vezes útil o acompanhamento do mentor.

Na segunda etapa, já integrado ao time ioasys, o trabalho em equipe e os clientes renomados requiriu um crescimento técnico fugaz, havendo feedbacks diários e o compartilhamento de boas práticas, elevando o nível de exigência, ocasionando em maior responsabilidade até então vivenciada por parte do estagiário

em sua carreira como desenvolvedor. Através de relatório do Bitbucket foi possível observar que houveram mais de 80 contribuições diretas ao longo dos projetos com clientes reais, além de 204 revisões de atividades de outros desenvolvedores.

A cultura adquirida durante todos os momentos na organização, a preocupação latente evidenciada pelo time de experiência e *people*, buscando o melhor estado da saúde mental e até física são elementos auxiliares que possuem mérito, pois implicaram diretamente na motivação do estagiário.

A mentoria é um programa que precisa passar por ajustes, tendo em vista que não era explicitado avaliações de desempenho e houve desatenção quanto a alinhamentos de expectativas. No que diz respeito a auxílio técnico, é inegável que a empresa possui profissionais altamente qualificados e com sentimentos altruístas, alinhado a disponibilidade foram recursos indispensáveis durante o estágio.

Por parte da direção da empresa, a transparência se mostrou presente através da realização de reuniões abertas com os donos da mesma ao final do ano, executando uma retrospectiva com números sobre faturamento, projetos entregues e reconhecimento das equipes. Também houve uma reunião no início do ano seguinte para explanar novos projetos de clientes, iniciativas e metas para o novo ciclo.

Em relação ao programa de estágio, é evidente que o impacto causado pela pandemia do COVID-19 ocasionou diversas situações as quais a empresa nunca passara, fazendo com que houvesse uma mudança drástica para o trabalho remoto de todos os times, o que resultou em que tomadas de decisões fossem realizadas erroneamente, pecando na comunicação. A lacuna pela falta de um plano de desenvolvimento individual é um desafio que persiste por parte de colaboradores na totalidade da empresa.

De maneira geral, a busca pela inovação e entregas com resultados surpreendentes são características que estiveram presentes no dia a dia da organização.

Por fim, o sentimento de gratidão é evidente a todas as experiências vivenciadas e compartilhadas no tempo de faculdade e empresa.



## REFERÊNCIAS

AGGARWAL, S. Modern web-development using reactjs. **International Journal of Recent Research Aspects**, v. 5, n. 1, p. 1–5, mar. 2018. Disponível em: <<http://ijrra.net/Vol5issue1/IJRRRA-05-01-27.pdf>>.

ATLASSIAN. **Breve visão geral do Bitbucket**. 2023. "Disponível em: "<<https://bitbucket.org/product/br/guides/getting-started/overview>>. Acesso em: 2 fev. 2023.

ATLASSIAN. **O que é Kanban?** 2023. "Disponível em: "<<https://www.atlassian.com/agile/kanban>>. Acesso em: 30 jan. 2023.

BALLARIN, G. **Como criar uma Landing Page que converte?** 2020. "Disponível em: "<<https://mestresdosite.com.br/blog/como-criar-landing-page>>. Acesso em: 1 fev. 2023.

BECK, K. et al. **Manifesto para Desenvolvimento Ágil de Software**. 2001. Disponível em: <<https://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 23 jan. 2023.

BOGNER, J.; MERKEL, M. To type or not to type? a systematic comparison of the software quality of javascript and typescript applications on github. **arXiv**, v. 1, n. 1, p. 1–12, mar. 2022. Disponível em: <<https://arxiv.org/abs/2203.11115>>.

CARVALHO, R. **Marketing digital: o que é, como funciona e estratégias infalíveis**. 2021. "Disponível em: "<<https://herospark.com/blog/marketing-digital>>. Acesso em: 1 fev. 2023.

KEMP, S. **DIGITAL 2022: GLOBAL OVERVIEW REPORT**. 2023. "Disponível em: "<<https://datareportal.com/reports/digital-2022-global-overview-report>>. Acesso em: 09 fev. 2023.

MATHEWS, K. **Gatsby - The future of web development is here**. 2023. "Disponível em: "<<https://github.com/gatsbyjs/gatsby>>. Acesso em: 15 jan. 2023.

META. **React - A JavaScript library for building user interfaces**. 2023. "Disponível em: "<<https://reactjs.org/>>. Acesso em: 15 jan. 2023.

MICROSOFT. **TypeScript for the New Programmer**. 2023. "Disponível em: "<<https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>>. Acesso em: 15 jan. 2023.

MICROSOFT. **Visual Studio Code - Overview**. 2023. "Disponível em: "<<https://code.visualstudio.com/docs>>. Acesso em: 7 fev. 2023.

MOZILLA. **CSS: Cascading Style Sheets**. 2023. "Disponível em: "<<https://developer.mozilla.org/en-US/docs/Web/CSS>>. Acesso em: 15 jan. 2023.

MOZILLA. **JavaScript**. 2023. "Disponível em: "<<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>. Acesso em: 15 jan. 2023.

MOZILLA. **Structuring the web with HTML**. 2023. "Disponível em: "<<https://developer.mozilla.org/en-US/docs/Learn/HTML>>. Acesso em: 15 jan. 2023.

SCRUM. **What is Scrum?** 2023. "Disponível em: "<<https://www.scrum.org/resources/what-is-scrum>>. Acesso em: 30 jan. 2023.

SINGHAL, P. **Frontend vs Backend**. 2023. "Disponível em: "<<https://www.geeksforgeeks.org/frontend-vs-backend/>>. Acesso em: 15 jan. 2023.

STACKOVERFLOW, S. **Web frameworks and technologies**. 2023. "Disponível em: "<<https://survey.stackoverflow.co/2022/>>. Acesso em: 09 fev. 2023.

STATCOUNTER. **Desktop vs Mobile Market Share Worldwide**. 2023. "Disponível em: "<<https://gs.statcounter.com/platform-market-share/desktop-mobile/worldwide/#yearly-2013-2022>>. Acesso em: 09 fev. 2023.

SUTHERLAND, J. **Scrum - a arte de fazer o dobro de trabalho na metade do tempo**. 1. ed. São Paulo: LeYa, 2014.

TEAMWORK. **Teamwork platform**. 2022. "Disponível em: "<<https://www.teamwork.com>>. Acesso em: 7 fev. 2023.