



LUÍS FELYPE FIORAVANTI FERREIRA MOREIRA

**DESENVOLVIMENTO DE SOFTWARE
LABORATORIAL PARA ANÁLISE DE SOLOS**

LAVRAS – MG

2021

LUÍS FELYPE FIORAVANTI FERREIRA MOREIRA

**DESENVOLVIMENTO DE SOFTWARE LABORATORIAL PARA
ANÁLISE DE SOLOS**

Relatório de estágio apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Graduação em Ciência da Computação, para a obtenção do título de bacharel.

Prof.^a Dr.^a Juliana Galvani Greggi
Orientadora

LAVRAS – MG

2021

LUÍS FELYPE FIORAVANTI FERREIRA MOREIRA

**DESENVOLVIMENTO DE SOFTWARE LABORATORIAL PARA
ANÁLISE DE SOLOS**

Relatório de estágio apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Graduação em Ciência da Computação, para a obtenção do título de bacharel.

APROVADA em 18 de Novembro de 2021.

Prof.^a Dr.^a Juliana Galvani Greghi UFLA
Prof. Dr. Joaquim Quinteiro Uchôa UFLA
Julio Cesar Sanches Corso Terras Gerais

Prof.^a Dr.^a Juliana Galvani Greghi
Orientadora

**LAVRAS – MG
2021**

Dedico este trabalho ao meu pai, Edmilson Vieira Gomes, e a minha mãe, Isabela Fioravanti Ferreira, por todo o amor e apoio, sem os quais a minha graduação e este trabalho não seria possível. Agradeço também ao meu avô, Ezequiel Manoel Ferreira, pelo exemplo de integridade e honestidade que me guiaram até aqui.

AGRADECIMENTOS

Agradeço primeiramente a empresa Terras Gerais, pela oportunidade e confiança de desenvolver este trabalho. Agradeço aos amigos que fizeram da jornada mais leve ao meu lado. Agradeço também a minha orientadora, Juliana Galvani Greggi, pela paciência, pelos ensinamentos, e pelas incontáveis horas de ajuda ao longo da minha graduação.

SIC PARVIS MAGNA (Sir Francis Drake)

RESUMO

O uso de ferramentas digitais na sociedade é cada vez mais expressivo. Dentre as várias áreas que tem se beneficiado dos avanços tecnológicos, a agricultura é uma das que utilizam sistemas computacionais para auxiliar suas atividades de modo a tornar o agronegócio cada vez mais competitivo. Um *software* para gestão de análises laboratoriais de solo pode beneficiar várias etapas da cadeia produtiva. Este trabalho apresenta um relato da experiência obtida durante a realização de estágio em uma empresa que atua no setor de serviços voltados ao agronegócio. Durante o estágio foi desenvolvido um *software* para gestão do laboratório de análises de solos. O resultado obtido foi satisfatório, proporcionando a otimização de tempo e mão-de-obra.

Palavras-chave: Web. Front-End. Agronegócio

ABSTRACT

The use of digital tools in society is increasingly expressive. Among the various areas that have benefited from technological advances, agriculture is one that uses computer systems for its auxiliary activities in order to make agribusiness increasingly competitive. A software for managing laboratory soil analysis can benefit several stages of the production chain. This work presents an account of the experience obtained during the internship in a company that operates in the agribusiness services sector. During the internship, a software was developed to manage the soil analysis laboratory. The result was satisfactory, providing the optimization of time and labor.

Keywords: Web. Front End. Agribusiness

SUMÁRIO

1	INTRODUÇÃO	15
2	REFERENCIAL TEÓRICO	19
2.1	A empresa Terras Gerais	19
2.2	Fundamentos teóricos	20
2.2.1	Planejamento e Desenvolvimento do Sistema	20
2.2.2	Usabilidade	22
2.2.3	Organização dos dados	23
2.3	Tecnologias e Ferramentas	24
2.3.1	PostgreSQL	24
2.3.2	GitHub	25
2.4	Python	25
2.5	HTML	26
2.6	CSS	26
2.7	Bootstrap	26
2.8	JavaScript	27
3	METODOLOGIA	29
3.1	Levantamento de requisitos	29
3.2	Planejamento e modelagem	30
3.3	Implementação e testes	31
4	RESULTADOS	33
4.1	Conhecendo o problema	33
4.2	Projetando e implementando a solução	33
5	CONCLUSÃO	41
	REFERÊNCIAS	43

1 INTRODUÇÃO

O uso de ferramentas digitais na sociedade é cada vez mais expressivo. Uma rápida análise das duas últimas décadas, nos mostra que o desenvolvimento de algumas tecnologias ultrapassou o seu objetivo primário de facilitar alguma atividade humana, vindo a se tornarem socialmente indispensáveis.

De acordo com James V McGee (MCGEE, 2004),

"A implementação do intercâmbio eletrônico de dados (EDI) para dar suporte ao intercâmbio de informações relativas a pedidos e faturamento, o intercâmbio de dados de projetos auxiliado por computador (*Computer Aided Design - CAD*) entre contratantes e subcontratantes durante o projeto de novos produtos, e a troca de correspondência eletrônica entre profissionais de empresas diferentes são exemplos iniciais de um ambiente geral de comércio em que todos os negócios rotineiros serão eletronicamente efetuados. Nos próximos anos, as operações comerciais eletrônicas serão tão comuns quanto são hoje as telefônicas".

Áreas como comunicação e propaganda se aproveitam de tecnologias que conectam pessoas para explorar o maior alcance possível na apresentação de ideias. Segundo Hobbs (HOBBS; RUSHKOFF, 2020), propagandas digitais tem sido utilizadas não apenas para fins comerciais, por exemplo, com líderes políticos utilizando tal estratégia para influenciar a opinião pública.

Na saúde, podem ser encontrados desde sistemas que agilizam um atendimento emergencial até sistemas de apoio à diagnóstico médico. Wu (WU, 2011) mostra que os sistemas de informação hospitalares podem ser utilizados para fornecer dados sobre pacientes, finanças, gestão executiva e várias outras áreas da cadeia hospitalar, permitindo que sejam extraídas informações que atendam as necessidades dos usuários nos mais variados setores, por exemplo, na tomada de decisões estratégica e outras necessidades apresentadas pelos usuários. Ainda sob esse mesmo enfoque, Aline Mapelli (MAPELLI; GIONGO; CARNEVALE, 2018) mostra que as informações extraídas podem ser ainda mais valiosas para a população se utilizadas, por exemplo, para o controle de epidemias e relata que o surto

de H1N1 pôde ser antecipado em várias regiões do mundo graças à aplicação de algoritmos específicos para a análise dos dados. A autora também relata que outras aplicações podem ter tanta relevância quanto a citada anteriormente, como a prevenção de desastres naturais ou na prevenção à criminalidade.

No entretenimento, Gilvam Vilarim (VILARIM, 2015) comenta como esta codificação digital conecta a um mesmo tecido eletrônico mídias como cinema, radiotelevisão, música, dentre outras, relegando a um segundo plano a questão do material.

Fica nítido, então, que o uso de sistemas da informação já está consolidado na sociedade, tornando-a mais rápida e dinâmica na realização de tarefas, sejam essas laborais ou não. E assim como as várias áreas citadas, o setor de Agronegócio tem evoluído muito com a utilização de sistemas computacionais diversos, desde a aplicação de maquinário inteligente até o controle logístico da produção. Os sistemas utilizados também são bastante diversos, podendo ser utilizados diretamente pelo produtor para gestão de suas propriedades, ou pelos profissionais da área para otimizar atividades relacionadas, por exemplo, à análise da fertilidade do solo ou controle de pragas (MASSRUHÁ S. M. F. S.; LEITE, 2017).

Todo esse avanço no uso de sistemas computacionais proporciona aos alunos do curso de Bacharelado em Ciência da Computação a possibilidade de exercer todo o conhecimento adquirido nas aulas, tanto teóricas quanto práticas, com a realização de estágios não-obrigatórios em empresas conveniadas a UFLA.

E é nesse contexto que este relatório de estágio apresenta o desenvolvimento de um sistema da informação construído sob demanda para o laboratório de análises químicas do solo, da empresa Terras Gerais, localizada em Lavras/MG, durante o período de março de 2020 até outubro do mesmo ano. O objetivo desse sistema foi aprimorar a coleta, armazenamento e processamento das informações obtidas nas análises químicas realizadas no laboratório. Aqui se encontram os registros sobre o processo de construção do software, desde a fase de planejamento

e coleta de requisitos, passando por implantações no laboratório até a finalização do estágio.

Este texto segue organizado da seguinte maneira: No cap. 2 serão apresentados os conceitos teóricos que embasaram o desenvolvimento do trabalho. O Cap. 3 mostra os processos empregados no desenvolvimento do sistema proposto. Os resultados obtidos serão descritos no Cap. 4 e, por fim, o Cap. 5 traz as conclusões e recomendações para a continuidade do trabalho.

2 REFERENCIAL TEÓRICO

Esta seção traz uma breve apresentação da empresa onde o trabalho foi realizado e os conceitos teóricos utilizados pelo autor para o desenvolvimento de seus trabalhos.

2.1 A empresa Terras Gerais

A Terras Gerais Consultoria¹ surgiu em 2009. Lidando com agricultura de precisão, a empresa se consolidou e tornou-se referência nacional. A Terras Gerais Consultoria oferece trabalhos de planejamento e acompanhamento de lavouras, realizando coleta de amostras do solo e elaboração de mapas de fertilidade e recomendações agronômicas. Mais tarde, em 2014, foi criada a Terras Gerais Experimental, com o intuito de operar na área de pesquisas agronômicas e de produtos nas lavouras, buscando sustentabilidade e produtividade.

Em 2020, a competitividade do agronegócio fez com que os responsáveis pela empresa pensassem em uma maneira de aumentar ainda mais sua produtividade interna, e nesse contexto, nasceu o projeto de desenvolvimento de um sistema para os seus laboratórios de análise química, no qual o autor trabalhou durante seu período de estágio.

Para o desenvolvimento do sistema, o autor teve a oportunidade de participar das decisões sobre a escolha da linguagem de programação a ser utilizada, do sistema de gerenciamento de bancos de dados, e outras escolhas necessárias, que serão descritas e fundamentadas na continuidade desse capítulo.

¹ Grupo Terras Gerais - Nossa História <<https://terrasgerais.com/o-grupo/>> - acessado em 26/10/2021

2.2 Fundamentos teóricos

Os conceitos teóricos aprendidos durante o curso foram empregados na realização de diversas atividades durante o período de estágio. Os principais conceitos serão apresentados nas próximas subseções.

2.2.1 Planejamento e Desenvolvimento do Sistema

A construção de um software não é trivial. A Engenharia de Software surgiu com o objetivo de parametrizar as fases que o desenvolvimento de um software deve ser submetido para que o resultado obtido no mínimo cumpra as funções propostas. Ao longo dos anos, várias metodologias de processo de software têm sido propostas, mas pode-se dizer que, em geral, um modelo de processo de software é composta por um conjunto de atividades principais e é complementada por uma série de atividades de apoio. As atividades principais podem ser descritas como (PRESSMAN ROGER. S.AND MAXIM, 2021):

1. Entender as necessidades do cliente para estabelecer os requisitos funcionais (funcionalidades disponíveis ao usuário) e não-funcionais (necessários, mas não necessariamente percebidos pelo cliente, por exemplo, requisitos de segurança dos dados);
2. Estabelecer as atividades técnicas a serem realizadas, identificar possíveis riscos de modo a minimizá-los, elaborar a documentação necessária para as próximas atividades e produzir artefatos indispensáveis, como o cronograma de trabalho;
3. Produzir os esboços (protótipos), modelos e diagramas que deverão nortear a construção do sistema;
4. Criar os códigos da aplicação e os testes a serem aplicados para a identificação de possíveis erros e falhas;

5. Entregar o sistema ao cliente, que passará a utilizá-lo e fornecerá *feedback* sobre possíveis incorreções ou novas demandas;

Vale ressaltar que a elaboração e aplicação dos testes não deve ser realizada apenas ao término da implementação do sistema, e sim ao longo de todo o processo (VAZQUEZ; SIMÕES, 2016).

Dentre as várias atividades de apoio possíveis, as seguintes atividades foram imprescindíveis para este projeto (PRESSMAN ROGER. S.AND MAXIM, 2021):

1. **Controle e acompanhamento do projeto:** permite que os envolvidos possam verificar a evolução do desenvolvimento e buscar soluções em caso de atrasos;
2. **Gerenciamento da Capacidade de reutilização:** estabelecer critérios e mecanismos para obtenção e reutilização de artefatos;
3. **Preparo e produção de artefatos:** atividades para elaboração de documentos, diagramas e outros artefatos necessários ao processo de desenvolvimento;

O aumento do uso de sistemas computacionais nas atividades cotidianas do ser humano levou ao aumento natural da complexidade envolvida no desenvolvimento de tais sistemas e, com a popularização do uso de sistemas web, uma nova divisão surgiu quando se pensa em tais sistemas: o *front-end* e o *back-end*. De maneira bastante simples, pode-se definir *front-end* como a porção do sistema que é enxergada e manipulada pelo usuário, é responsável pelos elementos vistos em tela e suas estilizações, e *back-end* pode ser entendido como a porção "central" do sistema, responsável pelas aplicações e dados desenvolvidos, e pela comunicação da aplicação com o banco de dados (SCHEIDT, 2015).

2.2.2 Usabilidade

Embora os sistemas computacionais sejam utilizados há várias décadas pelo homem, a forma de interação, a satisfação dos usuários, a manipulação, a experiência de uso e a maneira como as informações são apresentadas passou a figurar entre as preocupações das empresas de desenvolvimento há, relativamente, pouco tempo. Apenas quando o uso de sistemas computacionais tornou-se parte da vida cotidiana do ser humano é que os desenvolvedores e as empresas de desenvolvimento perceberam que a experiência do usuário pode influenciar diretamente na ampla aceitação e uso do sistema (BARBOSA; SILVA, 2010).

Ao longo dos anos surgiram várias abordagens para o desenvolvimento de interfaces que sejam adaptadas às necessidades e expectativas do usuário, e muitas das diretrizes criadas continuam sendo utilizadas (e as vezes atualizadas) mesmo após mais de uma década de sua criação. Um exemplo disso são as heurísticas² definidas por Nielsen (NIELSEN, 1994a) que servem como princípios gerais para o *desing* de interação e para a avaliação de interfaces (NIELSEN, 1994b). As 10 heurísticas são (tradução pelo autor):

1. Visibilidade do estado do sistema: o usuário deve ser informado do que está acontecendo com o sistema, recebendo *feedback* apropriado;
2. Compatibilidade entre o sistema e o mundo real: o sistema deve apresentar palavras, símbolos e conceitos que sejam familiares ao usuário mantendo as informações em ordem lógica;
3. Controle e liberdade para o usuário: o usuário deve ter a possibilidade de se recuperar de erros sem grande esforço;
4. Consistência e padronização: termos, conceitos e ações utilizados devem ser mantidos ao longo de todo o sistema;

² <http://nngroup.com/articles/ten-usability-heuristics>

5. Prevenção a erros: o sistema deve antecipar os possíveis erros cometidos, eliminando as situações sujeitas a erro ou fazendo verificações com o usuário ao longo do processo;
6. Reconhecimento no lugar de lembrança: o sistema deve manter ações, opções e outras informações visíveis ao usuário de modo a minimizar a carga de memorização necessária;
7. Flexibilidade e eficiência de uso: atalhos devem ser disponibilizados para usuários avançados (e ocultados para usuários novatos) de modo a permitir que o próprio usuário configure suas ações frequentes;
8. *Design* estético e minimalista: a interface não deve apresentar informações inúteis ou pouco utilizadas que podem competir, pela atenção do usuário, com as informações relevantes;
9. Ajuda ao usuário para reconhecer, diagnosticar e se recuperar de erros: mensagens de erro devem ser expressas em linguagem simples e devem indicar qual o problema e como o usuário pode corrigi-lo;
10. Ajuda e documentação: é melhor que o usuário não precise de ajuda para manipular o sistema, mas ainda assim, é necessário que informações que possam auxiliar o usuário na realização de suas tarefas sejam facilmente recuperadas;

Esse conjunto de heurísticas, quando usados para guiar o desenvolvimento, minimizam em muito a possibilidade de problemas que possam atrapalhar a interação do usuário com o sistema.

2.2.3 Organização dos dados

Uma das grandes vantagens da utilização de sistemas computacionais é a maior facilidade para a manipulação e armazenamento dos dados que, muitas ve-

zes, são massivos quantitativamente. Entretanto, para que estas vantagens possam ser observadas, é necessário que os dados sejam estruturados de maneira adequada, por exemplo, utilizando sistemas de gerenciamento de bancos de dados (SGBD) para seu armazenamento e manipulação.

Um banco de dados é uma coleção de dados devidamente relacionados. Já um Sistema de Gerenciamento de Bancos de Dados é definido como um sistema de software projetado para guardar e gerenciar bancos de dados (CARDOSO; CARDOSO, 2017). Um banco de dados relacional é um sistema que armazena os dados utilizando tabelas, estabelecendo relações diretas entre os mesmos, e a recuperação dos dados se dá através de operações relacionais (ELMASRI; NAVATHE, 2005).

Dentre as formas de organização de bancos de dados pode-se dizer que os relacionais figuram entre os mais utilizados. Um Bancos de Dados Relacional segue, em sua construção, um modelo baseado na teoria dos conjuntos. Além disso, é necessário que o banco de dados seja modelado seguindo as diretrizes de normalização, que são regras que ditam esta modelagem, a fim de manter o banco eficiente e sem redundâncias(MACHADO, 2018).

2.3 Tecnologias e Ferramentas

As atividades realizadas foram apoiadas por diversas ferramentas e tecnologias. Os conceitos principais acerca de cada uma delas será apresentado a seguir.

2.3.1 PostgreSQL

Buscando a maior robustez e confiabilidade possível dos dados a serem utilizados, o projeto utilizou o SGBD PostgreSQL. Segundo Stones e Matthew (STONES; MATTHEW, 2006) ele é uma excelente implementação de um banco de dados relacional, que disponibiliza uma grande gama de recursos, de código aberto, e de uso gratuito.

2.3.2 GitHub

À medida em que um software e a equipe trabalhando nele se expandem, difundir o desenvolvimento do sistema em versões para trabalho colaborativo é fundamental para não perder o controle tanto individual quanto coletivo do progresso do software. Para isso, é necessário uma ferramenta confiável, que hospede uma cópia do repositório a ser trabalhado, permita e controle o desenvolvimento com mais de um colaborador e mantenha versões do software, reversíveis a qualquer momento, se necessário. Assim, garante-se que o trabalho de um desenvolvedor não interfira no trabalho de outro, incorporando as alterações de cada um ao produto principal, sem desorganização.

O controle de versões pode ser considerado uma das atividades que integram o gerenciamento de mudanças, uma das atividades essenciais para a garantia de qualidade de software (PRESSMAN ROGER. S.AND MAXIM, 2021).

Embora existam diversas ferramentas disponíveis para realizar essa tarefa, Git é apontado como a mais popular desde 2016³ e essa popularidade se mantém até hoje^{4,5}. Dentro as várias plataformas *online* disponíveis para a utilização de Git, o GitHub figura entre as mais conhecidas⁶.

2.4 Python

Python é uma linguagem de programação interpretada que vem se tornando cada vez mais popular na indústria de tecnologia. Segundo Ramalho (RAMALHO, 2015), uma das características positivas da linguagem é a consistência, que permite que usuário consiga utilizar recursos e saiba quais são seus fundamentos sem sequer ter tido contato com os mesmos.

³ <https://rhodecode.com/insights/version-control-systems-2016>

⁴ <https://www.bairesdev.com/blog/which-version-control-system-developers-use/>

⁵ <https://www.softwaretestinghelp.com/version-control-software/>

⁶ <https://www.devopsschool.com/blog/top-5-git-hosting-solutions/>

2.5 HTML

Para se construir um sistema *web* de maneira adequada, é necessário que os elementos que compõem as páginas estejam dispostos de maneira organizada e satisfatória. Para essa tarefa, uma das possibilidades é o uso de HTML (*HyperText Markup Language*), uma linguagem de marcação de hipertexto utilizada para organizar e estruturar os conteúdos de uma página *web* ou *site*. Silva (SILVA, 2015) define Hipertexto como sendo todo texto inserido em um documento para a *web* e que tem como principal característica a possibilidade de se interligar a outros documentos da *web*. Todo conteúdo textual visto em uma página *web* é um hipertexto, e os elementos não textuais, em geral, são chamados de hiperímídia.

2.6 CSS

Embora HTML seja a linguagem escolhida para a estruturação e organização dos conteúdos, existem outras questões que devem ser consideradas para o desenvolvimento do sistema, por exemplo, a necessidade de estilização. A estilização das páginas será necessária? Caso a resposta seja positiva, qual tecnologia deve ser utilizada? O CSS (*Cascading Style Sheets*) é uma linguagem que possibilita a estilização de páginas *web* através do uso de folhas de estilo, artefato através do qual são definidas regras para a definição da aparência de uma página, por exemplo, a cor e localização dos elementos exibidos, ou os espaçamentos e margens utilizados (SCHEIDT, 2015).

2.7 Bootstrap

Com tantas tecnologias e linguagens diferentes, é fácil perder o controle dos códigos criados, dificultando sua manutenibilidade. Um *framework* é uma estrutura de códigos genérica que tem o objetivo de prover uma nova função dentro do código, se tornando muito útil para facilitar a construção de um sistema. O Bo-

Bootstrap 4 é um exemplo de um *framework*. Desenvolvido pelo Twitter, o objetivo principal do Bootstrap 4 é fornecer uma estrutura de *front-end* da *web* para desenvolvimento responsivo com compatibilidade entre navegadores (MORETO et al., 2017).

2.8 JavaScript

Foram apresentadas, até o momento, ferramentas e/ou tecnologias para a construção de um sistema *web* organizado, estruturado e estilizado. Mas o que se pode fazer dentro de uma página *web*?

Finalmente, para deixar o sistema interativo e interessante, foi utilizado JavaScript, que é uma linguagem de alto nível, dinâmica, interpretada e não tipada, conveniente para estilos de programação orientados a objetos e funcionais. Na opinião de Flanagan (FLANAGAN,), JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo de páginas Web; CSS, para especificar a apresentação dessas páginas; e JavaScript, para especificar seu comportamento.

3 METODOLOGIA

Para a construção do sistema de gestão da informação do laboratório de análises químicas da Terras Gerais, dividiu-se as tarefas em dois trabalhos distintos: este, que trata da criação do *front-end* do sistema, e outro responsável pelo *back-end*. Como visto anteriormente, o *front-end* é responsável pela apresentação das informações ao usuário. Além disso, é no *front-end* onde é tratada a interação entre as telas e seus elementos, e o usuário. Já o *back-end*, estabelece a comunicação entre a aplicação e o banco de dados, portanto é nele onde se constrói o modelo de requisições, tanto para envio como para recebimento de informações.

3.1 Levantamento de requisitos

Para transformar a necessidade da empresa em requisitos de software, eram realizadas reuniões de coleta de requisitos, aproximadamente a cada duas semanas, utilizando ambientes para reuniões virtuais (por exemplo, Google Meet) devido à pandemia de Covid-19, nas quais o cliente falava livremente sobre o que precisava. Para tentar resolver possíveis omissões involuntárias, também eram feitas perguntas para melhor entender o processo de trabalho e domínio dos dados que deveriam ser tratados pelo sistema. Além das funcionalidades, várias informações sobre os requisitos não funcionais e sobre a expectativa de uso do sistema foram coletadas para dar suporte a todo o processo de desenvolvimento, inclusive, da interface de interação e do banco de dados a ser utilizado para armazenamento e consulta dos dados relacionados às análises do laboratório. Ocorreram algumas alterações dos requisitos ao longo do processo de desenvolvimento e, embora seja comum que o cliente faça alterações nos requisitos, os processos a serem implantados no laboratório de análises de solo ainda não estavam suficientemente consolidados e, por isso, várias alterações foram necessárias.

3.2 Planejamento e modelagem

Para que o desenvolvimento não sofresse com retrabalho ocasionado por requisitos não atendidos, e nem com correções por funcionalidades não contempladas, tanto a aplicação em si quanto a estrutura de armazenamento e manipulação dos dados foram planejados previamente. O planejamento da aplicação pôde ser dividido em dois momentos: planejamento das interfaces de interação com o usuário e planejamento das funcionalidades disponíveis. As interfaces foram planejadas, inicialmente, em protótipos de baixa fidelidade, sugeridos pela orientadora do estágio. Esses primeiros esboços eram discutidos e decidia-se como os elementos que compunham as telas deveriam estar dispostos nas mesmas. As telas eram criadas, primeiramente seguindo os protótipos, e quando terminadas, as telas eram testadas de duas maneiras: eram relidos todos os requisitos da tela finalizada, em formato de tópicos, objetivando garantir que todos estes requisitos foram englobados. Já o segundo teste foi seguir o fluxo do processo requisitado pela empresa, garantindo sua execução sem falhas. Pode-se dizer que o desenvolvimento dos protótipos de alta fidelidade foi incremental uma vez que, após várias rodadas de avaliação e correção, as telas que eram protótipos continuaram a ser utilizadas como telas oficiais. O projeto tinha por objetivo a construção de um software laboral, e, nesse caso, quanto melhor a experiência do usuário com a utilização do software, mais confortável ele se sentiria para realizar suas tarefas diariamente. Para isso, o planejamento das interfaces levou em consideração as heurísticas de Nielsen (NIELSEN, 1994b) e várias dela, como o oferecimento de *feedback* de ações no software, as confirmações destas ações, e a prevenção a erros, podem ser observadas nas telas desenvolvidas, apresentadas no Cap. 4.

Como relação ao planejamento do sistema, uma das decisões mais importantes foi a de que seria utilizada a componentização dos elementos do sistema visando aumentar o reuso de código e a facilidade de manutenção. Outra decisão

realizada na fase de planejamento do sistema foi a definição de uso da linguagem Python uma vez que a demanda do cliente era por um sistema *desktop*.

Para o armazenamento e manipulação dos dados foi escolhido um SGBD Relacional e, para guiar a implementação do banco de dados, as possíveis tabelas foram modeladas a partir dos dados coletados. As tabelas foram modeladas e como já relatado, ocorreram várias mudanças no conjunto de requisitos e várias delas afetaram diretamente a modelagem criada. Dessa forma, o planejamento inicial sofreu várias alterações até que se alcançasse estabilidade nos requisitos definidos e o modelo pudesse ser implementado sem que houvesse necessidade de alterações mais relevantes, levando a grande esforço e retrabalho. A seguir, são apresentadas algumas das tabelas modeladas (Figura 3.1).

Figura 3.1 – Possíveis tabelas a serem implementadas

```
Pessoa
[ idPessoa, CpfCnpj, InscricaoEstadual, Nome, Telefone, Email]

Endereco
[Tipo, Rua, Numero, Complemento, CEP, Cidade, Estado, Pais]

Propriedade
[idPropriedade, Inscr.Produutor, NomePropriedade, Endereco, Cidade, Estado, Pais]

Talhao
[idTalhao, NomeTalhao]

Amostra
[idAmostra]

Lote
[idLote, DataLote, QuantidadeAmostras]
```

Após a primeira modelagem do banco, foi gerado o repositório responsável pelo controle de versões do código o processo de criação do software. Nesse ponto, já estavam disponíveis as ferramentas selecionadas para que começasse, de fato, a criação das telas do sistema.

3.3 Implementação e testes

Com a decisão de que o sistema desenvolvido seria *desktop* e com os protótipos das telas, o banco de dados modelados e o repositório para controle de ver-

sões criado, foi iniciado o processo de desenvolvimento do sistema propriamente dito. Como mencionado anteriormente, o desenvolvimento completo do sistema se deu em duas frentes de trabalho e o autor deste relatório atuou principalmente no desenvolvimento do *front-end*.

Devido à experiência prévia do autor, a linguagem escolhida foi Python. O desenvolvimento foi iniciado e seu uso foi extremamente útil. Entretanto, uma das mudanças mais relevantes promovida ao longo do processo foi o de plataforma: o sistema passaria a ser utilizado via *web* e, a partir daí, o uso da linguagem Python não era mais viável. Isso ocorreu devido a incompatibilidades entre versões de bibliotecas necessárias para a execução do programa nos laboratórios. Para o desenvolvimento do sistema para *web* passaram a ser utilizados HTML5, JavaScript, CSS3 e Bootstrap4. Outras informações sobre as mudanças e a continuidade da implementação serão apresentados no Cap. 4.

Ao longo de todo o projeto eram realizadas reuniões, em média, a cada duas semanas para discutir o que tinha sido feito até o momento e decidir as próximas prioridades.

4 RESULTADOS

O estágio aqui relatado foi realizado entre os meses de março e outubro de 2020. No final de março de 2020 foi dado início ao trabalho com a primeira reunião de coleta de requisitos. Nesta reunião foram definidos, além dos requisitos iniciais, as duas primeiras macro-tarefas do laboratório que o sistema deveria ser capaz de gerenciar: análise de solos e análise de nematoides, com prioridade na primeira. A função de análise de solos deveria englobar 3 tipos de análises: titulometria, absorção atômica e espectrofotometria. Já a análise de nematoides deveria ser realizada de maneira atômica, com a geração de 2 relatórios, divididos em análise nematológica do solo e de raízes, e o relatório de sementes.

4.1 Conhecendo o problema

Como já informado, os requisitos foram obtidos a partir de reuniões virtuais, através das quais as informações sobre as necessidades e expectativas da empresa puderam ser conhecidas. Além dessas informações, conhecer o processo de trabalho durante a análise de solos era fundamental para que o fluxo de informações no sistema pudesse ser estabelecido de maneira adequada. Como esse processo ainda não estava totalmente consolidado, várias alterações foram necessárias até que o fluxo de informações pudesse ser conhecido e o modelo de interação pudesse ser definido.

4.2 Projetando e implementando a solução

A modelagem do banco de dados e o planejamento do sistema puderam ser estruturados a partir dos requisitos colhidos. O banco de dados foi modelado, em seguida, as tabelas foram definidas. Com as tabelas definidas, visando a otimização da estrutura, o estabelecimento das relações entre as tabelas e a eliminação da redundância, foi aplicado o processo de normalização, sendo que a versão fi-

nal, disponível no momento em que o estagiário encerrou suas atividades junto a empresa, encontrava-se em conformidade com a terceira forma normal.

Neste projeto, o SGBD relacional utilizado foi o PostgreSQL. Presente do início ao fim do estágio, o PostgreSQL supriu a demanda do sistema e foi capaz de suportar a aplicação, considerando performance, expansão, e manutenibilidade do banco de dados. Uma vez modelado e normalizado, o banco foi criado dentro do SGBD, estando assim pronto ser utilizado pela aplicação principal.

Inicialmente, foi pedido pela Terras Gerais um *software desktop*. Pensando nisso, o Python 3 foi utilizado para construir telas e interações do sistema. Os elementos do *software* como campos de texto e rótulos eram todos providenciados pela biblioteca de interface gráfica do Python3, chamada Tkinter, bem como sua estilização. A única exceção desta estilização foram os botões. As imagens que representavam os botões eram criadas através de um gerador de imagens gratuito e inseridas no código com o intuito de deixar sua função mais intuitiva.

Ressalta-se que para cada botão, foi necessária a criação de duas imagens, uma para a representação do botão em seu estado normal, e outra para o estado pressionado.

Para satisfazer as necessidades relacionadas ao processo realizado dentro do laboratório de análise de solo, a navegação do software foi construída através de um sistema de abas, localizada na parte superior da tela, permitindo que o usuário pudesse navegar facilmente entre os tipos de análise. Essa característica pode ser percebida na imagem apresentada na Figura 4.1.

Devido ao pedido de prioridade feito pela empresa, a próxima etapa do trabalho foi a priorização da construção das telas relacionadas aos diversos tipos de análise de solos. Primeiramente, foi construída a tela de titulometria, que contava com uma caixa de seleção, e um local de exibição da tabelas, além dos botões de fechar e confirmar. O local de exibição de tabelas foi definido pois havia a necessidade de apresentação dos dados disponíveis em tabelas diferentes. Nesse

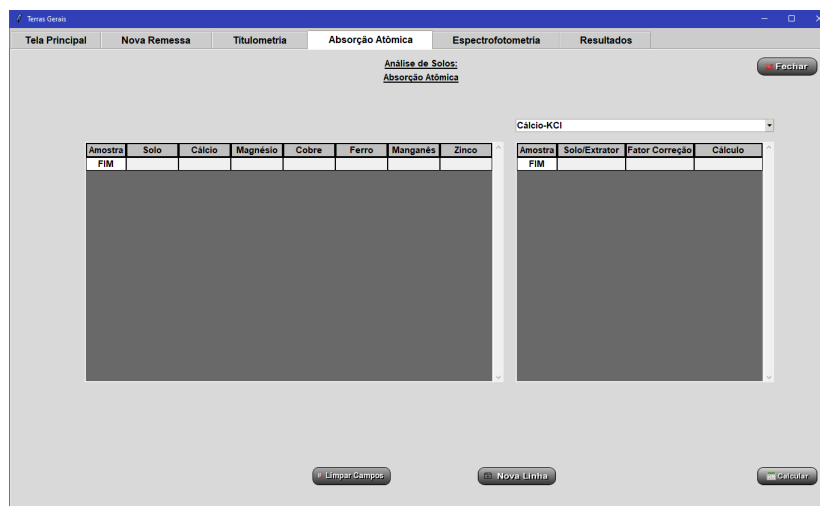
Figura 4.1 – Tela Inicial



local, poderiam ser exibidos os dados de de 7 tabelas, e a caixa de seleção controlava qual tabela estava sendo exibida no momento. Neste momento, o número de linhas e colunas destas tabelas eram fixos. Uma vez inseridos os dados de entrada nas tabelas, o botão "confirmar" deveria ser pressionado para que o *software* realizasse os cálculos exigidos pela análise e pudessem ser armazenasse no banco de dados. As telas de absorção atômica (Figura 4.2) e espectrofotometria eram bem similares à de titulometria, se diferenciando apenas no cálculo matemático, e especificações das tabelas em si. A tela de espectrofotometria, em especial, possuía uma tabela fixa que servia de consulta ao técnico do laboratório.

No fim do mês de abril, foi realizada uma nova reunião com *stakeholders*, ou seja, os profissionais da empresa diretamente ligados ao sistema, para validar o que tinha sido produzido até o momento e para estabelecer as próximas prioridades. Nesta reunião foi definida a terceira macro-tarefa: o controle de clientes e suas fazendas, das quais vêm as amostras que são analisadas pelo laboratório.

Figura 4.2 – Tela de Absorção Atômica em Python



Além disso, também foi pedido que um sistema de *login* de usuários fosse integrado ao sistema.

O estagiário começou, então, a trabalhar no processo de login. Seguindo as diretrizes de usabilidade, o processo de login foi proposto em duas telas: a tela inicial apresenta dois campos, nome de usuário e senha, e os botões para prosseguir no processo de autorização ou cancelar. Os dados preenchidos são verificados e, em caso de sucesso, é concedido o acesso às outras funções do software. A segunda tela proposta permite o cadastro de usuários que ainda não tenham acesso ao sistema.

Para atender integralmente às demandas apresentadas pelo cliente na última reunião, o estagiário passou a trabalhar na porção do sistema voltada a manipulação dos dados dos clientes. A partir da implementação dessa funcionalidade, as amostras de solo que iriam ou para análise de nematoides ou para análise de solos, passa a contar com as informações do cliente responsável por elas e da fazenda a qual pertencem. Uma nova tela foi produzida com o intuito de cadastrar as informações do cliente, de suas fazendas e seus talhões, porções demarcadas de terreno da fazenda de onde são retiradas as amostras enviadas ao laboratório.

Uma nova reunião foi realizada e novas demandas foram apresentadas pela empresa. Entretanto, dessa vez, as demandas tiveram maior impacto no trabalho já realizado: o processo de análise do laboratório foi alterado, afetando o fluxo de utilização do software. A análise de solos passou a necessitar de uma tela de cadastro de remessas, que passaram a ser utilizadas nas outras funcionalidades do sistema. Essa funcionalidade de remessas, em especial, teve seu desenvolvimento um pouco prolongado, pois houveram algumas mudanças quanto às remessas, que são o agrupamento de amostras. Uma vez finalizada, foi feita a entrega da primeira versão do sistema com todas as funcionalidades finalizadas até o momento. Toda essa etapa de desenvolvimento para a entrega da primeira versão do sistema durou cerca de 4 meses.

Ocorreram incompatibilidades com a versão do Python3 disponível na empresa, e a atualização através desse processo para entregar novas funcionalidades e versões seria muito árdua. Após a análise dos pontos positivos e negativos, e dos riscos e dificuldades que poderiam ser enfrentados no futuro, foi decidido que a plataforma do sistema seria trocada e o sistema passaria a ser um sistema *web*. Essa, na opinião do estagiário, foi a melhor decisão de projeto de todo o estágio. Por sua experiência, o autor acredita que o desenvolvimento de um sistema *web* é mais rápido, desde que sejam utilizadas as tecnologias adequadas, mais versátil pela sua facilidade em incorporar alterações e componentizar elementos, e pode desfrutar de tecnologias consolidadas e em constante aprimoramento.

Para dar início aos trabalhos de desenvolvimento do sistema na nova plataforma, foram escolhidas as linguagens HTML, CSS e Javascript para a elaboração do conteúdo e estilização das páginas, e o Bootstrap 4 para o aprimoramento da estética do software mais agradável.

Todas as funções existentes no *software desktop* foram rapidamente recriadas no novo sistema web, e o resultado foi tão bom quanto o esperado. Um exemplo disso pode ser visto nas Figuras 4.3 e 4.4, em que estão apresentadas a

tela de titulometria desenvolvida no sistema *desktop* e desenvolvida para o sistema *web*, respectivamente.

Figura 4.3 – Tela de Titulometria em Python

Figura 4.4 – Tela de Titulometria Web

Os principais elementos vistos em tela, como tabelas e caixa de seleção, foram componentizados, graças à versatilidade do Javascript. Isto implicou na

criação e reuso de código para esses elementos e suas interações, ocasionando uma diminuição considerável no tempo de produção.

A partir desse momento, as entregas para a empresa foram feitas de maneira remota, sem a necessidade ir presencialmente à empresa para aplicar novas funcionalidades.

Quando o sistema *web* alcançou o mesmo ponto de desenvolvimento que o sistema *desktop* apresentava quando foi substituído, foram solicitadas alterações nos requisitos do banco de dados, alterando a base de formação do software. Os requisitos sofreram alterações consideráveis, ocasionando mudanças na estrutura do banco e, conseqüentemente, nos elementos vistos nas interfaces de interação.

Após a incorporação destas alterações, uma nova demanda foi apresentada pela empresa: o desenvolvimento das interfaces de interação para a análise nematológica. O processo estabelecido foi bastante simples, sendo necessário apenas a apresentação de duas tabelas, Raízes e Solos, e o desenvolvimento de dois relatórios: uma para raízes e outro para solos. A Figura 4.5 apresenta um exemplo de relatório gerado pelo sistema.

Figura 4.5 – Relatório de Raízes

Laboratório de Nematologia e Fitopatologia					
Terras Gerais Experimental					
www.terrasgerais.com					
Lavras/MG					
Laudo análise de nematóides					
Nome Proprietário:					
Propriedade:					
Município/ Estado coleta:					
Responsável coleta:					
Cultura:					
Data amostragem:			Total Área Plantada:		
Data entrada no laboratório:			Contato/E-mail: Rafael Tadeu		
			(35)99875-8334		
			rafael.gouro@gmail.com		
Quantidade de Fitonematóides na amostra					
*Amostra	Espécies	Ovos/g raiz	J2/g raiz	Espécimes/g raiz	200 cm ³ solo
Amostra	Ovos	-	-	-	-
	Meloidogyne sp.	-	-	-	-
	Pratylenchus brachyusus	-	-	-	-
	Helicotylenchus	-	-	-	-
	Tylenchulus semipenetrans	-	-	-	-

A última atividade realizada pelo estagiário foi a hierarquização de usuários. Os usuários do sistema passaram a ser categorizados de acordo com um

conjunto de papéis e permissões de acesso distintos: administrador, técnico solos, técnico nematoide, técnico gerencial, estagiário solos, estagiário nematoide e estagiário gerencial. O fim do estágio se deu com a criação dessas posições para cada usuário, mas a influência de cada posição dentro do *software* não foi implementada. O aluno trabalhou na implementação do sistema *web* por cerca de 3 meses, até o término do estágio.

Mesmo com a finalização do estágio, o *software* continuou a ser desenvolvido dentro da Terras Gerais, com outra equipe. O estágio gerou uma experiência única para o estagiário, agregando conhecimento e maturidade profissional, não só pela parte técnica de desenvolvimento de um *software*, mas pelo aprendizado de trabalho em equipe, responsabilidades, metas e prazos. Com esta experiência, foi possível ter o contato com a realidade empresarial, que era o objetivo definido pelo estagiário.

5 CONCLUSÃO

O objetivo deste documento foi apresentar o desenvolvimento do *front-end* de um sistema laboratorial de análise de solos. As metas definidas para o projeto foram concluídas até o fim o estágio. Embora a finalização do estágio não tenha coincidido com o encerramento da construção do sistema, o trabalho apresentado gerou uma versão estável avaliada como satisfatória pela Terras Gerais.

Foram utilizados conhecimentos obtidos nas disciplinas de Engenharia de Software, Interação Humano-Computador, Estrutura de Dados, Introdução a Banco de Dados, Sistemas Gerenciadores de Banco de Dados e Desenvolvimento *Web*. A bagagem teórica obtida durante o curso foi ideal para se pensar na resolução de problemas e considerar diferentes abordagens, enquanto as ações práticas que o curso oferece são uma pequena dosagem das situações vividas no meio profissional.

REFERÊNCIAS

- BARBOSA, S. D. J.; SILVA, B. S. **Interação Humano-Computador**. [S.l.]: Elsevier, 2010. (Sociedade Brasileira de Computação). ISBN 9788535234183.
- CARDOSO, G.; CARDOSO, V. **SISTEMA DE BANCO DE DADOS**. Saraiva Educação S.A., 2017. ISBN 9788502162846. Disponível em: <<https://books.google.com.br/books?id=YYNnDwAAQBAJ>>.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de Bancos de Dados**. [S.l.]: Addison Wesley., 2005. ISBN 8588639173.
- FLANAGAN, D. **JavaScript: O Guia Definitivo**. Bookman Editora. ISBN 9788565837484. Disponível em: <<https://books.google.com.br/books?id=zWNYDgAAQBAJ>>.
- HOBBS, R.; RUSHKOFF, D. **Mind Over Media: Propaganda Education for a Digital Age**. W. W. Norton, 2020. ISBN 9780393713510. Disponível em: <<https://books.google.com.br/books?id=gC3xDwAAQBAJ>>.
- MACHADO, F. **Banco de Dados - Projeto e Implementação**. Saraiva Educação S.A., 2018. ISBN 9788536509846. Disponível em: <<https://books.google.com.br/books?id=N4diDwAAQBAJ>>.
- MAPELLI, A.; GIONGO, M.; CARNEVALE, R. **Os impactos das novas tecnologias no Direito e na Sociedade**. Deviant, 2018. ISBN 9788553240258. Disponível em: <https://books.google.com.br/books?id=UJx_DwAAQBAJ>.
- MASSRUHÁ S. M. F. S.; LEITE, M. A. d. A. Agro 4.0 - rumo à agricultura digital. In: **JC na Escola Ciência, Tecnologia e Sociedade: Mobilizar o Conhecimento para Alimentar o Brasil**. Embrapa Agricultura Digital, 2017. p. 28–35. Disponível em: <<http://www.alice.cnptia.embrapa.br/alice/handle/doc/1073150>>.
- MCGEE, J. **Gerenciamento estratégico da informação**. Campus, 2004. (Gerenciamento da Informação). ISBN 9788570019240. Disponível em: <<https://books.google.com.br/books?id=WwHvCKH3hS0C>>.
- MORETO, S. et al. **Bootstrap 4 – Responsive Web Design**. Packt Publishing, 2017. ISBN 9781788395274. Disponível em: <<https://books.google.com.br/books?id=MedDDwAAQBAJ>>.
- NIELSEN, J. Enhancing the explanatory power of usability heuristics. In: **Proceedings of the ACM CHI'94 Conf**. Boston, MA: ACM, 1994. p. 152–158.
- NIELSEN, J. Heuristic evaluation. In: NIELSEN, J.; MACK, R. (Ed.). **Usability Inspection Methods**. [S.l.]: John Wiley & Sons, 1994.

PRESSMAN ROGER. S.AND MAXIM, B. R. **Engenharia de Software - Uma abordagem profissional**. AMGH Editora Ltda., 2021. ISBN 9781259872976 / 1259872971. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9786558040118/>>.

RAMALHO, L. **Python fluente: Programação clara, concisa e eficaz**. Novatec Editora, 2015. ISBN 9788575224625. Disponível em: <<https://books.google.com.br/books?id=XqbfCgAAQBAJ>>.

SCHEIDT, F. **Fundamentos de CSS: criando design para sistemas web**. Outbox Livros Digitais, 2015. ISBN 9788566664041. Disponível em: <<https://books.google.com.br/books?id=04cbCgAAQBAJ>>.

SILVA, M. **Fundamentos de HTML5 e CSS3**. Novatec Editora, 2015. ISBN 9788575224380. Disponível em: <<https://books.google.com.br/books?id=2iPYCQAAQBAJ>>.

STONES, R.; MATTHEW, N. **Beginning Databases with PostgreSQL: From Novice to Professional**. Apress, 2006. (Books for professionals by professionals). ISBN 9781430200185. Disponível em: <<https://books.google.com.br/books?id=Nc5ZT2X5mOcC>>.

VAZQUEZ, C.; SIMÕES, G. **Engenharia de Requisitos: software orientado ao negócio**. Brasport, 2016. ISBN 9788574527901. Disponível em: <<https://books.google.com.br/books?id=gA7kDAAAQBAJ>>.

VILARIM, G. **Trabalho E Software No Capitalismo Cognitivo**. [s.n.], 2015. Disponível em: <<https://books.google.com.br/books?id=uPlxDwAAQBAJ>>.

WU, Y. **Computing and Intelligent Systems: International Conference, ICCIC 2011, held in Wuhan, China, September 17-18, 2011. Proceedings**. Springer Berlin Heidelberg, 2011. (Communications in Computer and Information Science, pt. 4). ISBN 9783642240904. Disponível em: <<https://books.google.com.br/books?id=-uuWJBAaoTAC>>.