



**LUIS HENRIQUE SAES SCANDELAI**

**RELATÓRIO DE ESTÁGIO:  
DESENVOLVIMENTO E MANUTENÇÃO DE  
SOFTWARE NA EMPRESA EMBRAER**

**LAVRAS – MG**

**2023**

**LUIS HENRIQUE SAES SCANDELAI**

**RELATÓRIO DE ESTÁGIO: DESENVOLVIMENTO E MANUTENÇÃO DE  
SOFTWARE NA EMPRESA EMBRAER**

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

Prof. Dr. Antônio Maria Pereira de Resende  
Orientador

**LAVRAS – MG**

**2023**

**Ficha catalográfica elaborada pela Coordenadoria de Processos Técnicos  
da Biblioteca Universitária da UFLA**

Scandelai, Luis Henrique Saes

Relatório de Estágio: Desenvolvimento e Manutenção de Software na Empresa Embraer / Luis Henrique Saes Scandelai. 1<sup>a</sup> ed. rev., atual. e ampl. – Lavras : UFLA, 2023.

42 p. : il.

Relatório de Estágio(bacharelado)–Universidade Federal de Lavras, 2023.

Orientador: Prof. Dr. Antônio Maria Pereira de Resende.

Bibliografia.

1. TCC. 2. Monografia. 3. Dissertação. 4. Tese. 5. Trabalho Científico – Normas. I. Universidade Federal de Lavras. II. Título.

CDD-808.066

**LUIS HENRIQUE SAES SCANDELAI**

**RELATÓRIO DE ESTÁGIO: DESENVOLVIMENTO E MANUTENÇÃO DE  
SOFTWARE NA EMPRESA EMBRAER**

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

APROVADA em 27 de Outubro de 2023.

Dr. Paulo Afonso Parreira Júnior UFLA

Dra. Paula Christina Figueira Cardoso UFLA

Prof. Dr. Antônio Maria Pereira de Resende  
Orientador

**LAVRAS – MG  
2023**

*À minha mãe, Marli Saes, por sempre estar ao meu lado, me apoiar em minhas  
decisões e ser a maior referência da minha vida.*

*Ao meu pai, José Roberto Scandelai, por ser um exemplo de pai e homem e por  
me ensinar que o trabalho duro é recompensador.*

*Aos meus amigos Caio Marchiorato, Daniel Tavares, Giovana Beraldo, Gustavo  
de Freitas, Igor Zaboroski, Marlene Saes, Miguel Oliveira e família, pelo apoio.*

*Dedico*

## **AGRADECIMENTOS**

À Universidade Federal de Lavras, em especial ao Departamento de Ciência da Computação, pelos ensinamentos.

À Embraer, pela oportunidade de estágio.

Ao professor Dr. Paulo Afonso Parreira Júnior, pela orientação, paciência, disposição e vontade de ajudar.

A todos meus amigos e colegas de curso que tornaram essa jornada mais prazerosa. A todos funcionários da UFLA.

Aos meus pais, pelo apoio e suporte.

*“Embora seja sempre bom acreditar em si mesmo, uma ajudinha dos outros  
pode ser uma grande benção.”  
(Iroh - Avatar: A Lenda de Aang)*

## RESUMO

Pretende-se, neste relatório de estágio, descrever as atividades de desenvolvimento de software, manutenção do código existente, automação de processos e junção de bases de dados para gestão da construção de múltiplos modelos de aeronaves, realizado na empresa Embraer. Dentro do contexto da empresa, os sistemas de software mantidos e aprimorados incluem o *Cockpit*, responsável pela administração de múltiplas fases dos processos de construção da aeronave, revisão direta com os clientes e geração de relatórios exportáveis e o *Weighing*, responsável pela administração dos processos de pesagem e auditoria da qualidade atual da aeronave. Portanto, no relatório, estão apresentadas as descrições das atividades desenvolvidas, bem como as tecnologias e metodologias empregadas durante o estágio. Como conclusão do período de estágio, destaca-se que este aprimorou o repertório técnico, suas qualificações pessoais e técnicas do estagiário, preparando-o para desafios no seu futuro profissional, também permitindo a ele praticar e consolidar habilidades de trabalho em equipe, desenvolvimento, manutenção e evolução de software e aplicar conceitos aprendidos em sala de aula.

**Palavras-chave:** Desenvolvimento Web. Engenharia de Software. Desenvolvimento Ágil de Software.



## ABSTRACT

The aim of this training report is to describe the software development activities, maintenance of existing code, process automation and joining of databases to manage the construction of multiple aircraft models, carried out at the company Embraer. Within the company context, software systems maintained and improved include *Cockpit*, responsible for managing multiple phases of aircraft construction processes, direct review with customers and generating exportable reports, and *Weighing*, responsible for managing the weighing processes and auditing the current quality of the aircraft. Therefore, the report presents descriptions of the activities carried out, as well as the technologies and methodologies used during the internship. As a conclusion to the internship period, it is noteworthy that it improved the intern's technical repertoire, personal and technical qualifications, preparing him for challenges in his professional future, also allowing him to practice and consolidate teamwork skills, development, software maintenance and evolution and apply concepts learned in the classroom.

**Keywords:** Web Development. Software Engineering. Agile Software Development.

## LISTA DE FIGURAS

Figura 2.1 – Quadro kanban digital . . . . .	16
Figura 2.2 – Processo scrum . . . . .	19
Figura 2.3 – Resultado da adição de CSS a um código base ASP.NET . . . .	20
Figura 2.4 – Exemplo de código CSS . . . . .	20
Figura 2.5 – Exemplo de consulta Oracle SQL . . . . .	21
Figura 2.6 – Exemplo de código Python para renomear arquivos . . . . .	22
Figura 2.7 – Seção de tarefas em aberto Jira . . . . .	23
Figura 3.1 – Fluxo para analisar e unir bases de dados distintas . . . . .	28
Figura 3.2 – Planilha “DE-PARA” . . . . .	29
Figura 3.3 – Consultas SQL para criação de tabelas de exemplo <i>foods</i> . . . .	30
Figura 3.4 – Consultas SQL para criação de tabelas de exemplo <i>dishes</i> . . . .	31
Figura 3.5 – Adiciona nova coluna OLD_ID na tabela <i>foods1</i> . . . . .	31
Figura 3.6 – Adiciona os registros na tabela <i>foods1</i> sem conflito . . . . .	32
Figura 3.7 – Adiciona os registros na tabela <i>foods1</i> com conflito . . . . .	32
Figura 3.8 – Adiciona nova coluna FROM_DISHES2 na tabela <i>dishes1</i> . . . .	32
Figura 3.9 – Adiciona os registros na tabela <i>dishes1</i> sem conflito . . . . .	32
Figura 3.10 – Adiciona os registros na tabela <i>dishes1</i> com conflito . . . . .	33
Figura 3.11 – Atualiza a referência da chave estrangeira . . . . .	33
Figura 3.12 – Remove colunas desnecessárias . . . . .	34
Figura 3.13 – Consulta SQL para validação dos registros . . . . .	35
Figura 3.14 – Código python para renomear arquivos e mover entre diferen- tes locais . . . . .	36
Figura 3.15 – Código python para copiar arquivos de uma pasta para outra ignorando cópias . . . . .	38

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	11
<b>1.1</b>	<b>Contextualização</b>	11
<b>1.2</b>	<b>Sobre a Embraer</b>	12
<b>1.3</b>	<b>Objetivos</b>	13
<b>1.4</b>	<b>Estrutura do trabalho</b>	14
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	15
<b>2.1</b>	<b>Kanban Board</b>	15
<b>2.2</b>	<b>Scrum</b>	17
<b>2.3</b>	<b>Tecnologias utilizadas</b>	19
<b>2.3.1</b>	<b>CSS</b>	19
<b>2.4</b>	<b>Banco de dados</b>	20
<b>2.5</b>	<b>Python</b>	21
<b>2.6</b>	<b>Gestão de Projetos com <i>Jira</i></b>	23
<b>3</b>	<b>ATIVIDADES REALIZADAS</b>	25
<b>3.1</b>	<b>Descrição das atividades admissionais</b>	25
<b>3.2</b>	<b>Descrição das atividades desenvolvidas no <i>Cockpit</i></b>	26
<b>3.2.1</b>	<b>Junções das tabelas da base de dados da qualidade</b>	26
<b>3.2.2</b>	<b>Atualização da nomeação de arquivos para realizar junções de servidores</b>	35
<b>3.2.3</b>	<b>Atualização da estrutura dos menus</b>	36
<b>3.3</b>	<b>Descrição das atividades desenvolvidas no <i>Weighing</i></b>	37
<b>3.3.1</b>	<b>Junções das tabelas da base de dados da qualidade</b>	37
<b>3.3.2</b>	<b>Correções relativas à alterações manuais nas bases de dados</b>	37
<b>4</b>	<b>CONCLUSÃO</b>	39
	<b>REFERÊNCIAS</b>	41

## 1 INTRODUÇÃO

Neste capítulo, é apresentada uma breve descrição da empresa Embraer, assim como algumas das dificuldades tratadas durante esse estágio supervisionado.

### 1.1 Contextualização

Durante a pandemia, um grande número de empresas brasileiras teve que se adaptar ao modelo do trabalho remoto. De acordo com a pesquisa “Gestão de Pessoas na Crise COVID-19”, elaborada pela Fundação Instituto de Administração (FIA), no ano de 2020, 46% das empresas nacionais adotaram o *home-office* permanente e a antecipação de férias (XTRATEGIE FIA, 2023).

Esse aumento expressivo na quantidade de empresas que adotaram o *home-office* permanente permitiu que pessoas de diferentes regiões pudessem procurar oportunidades em outros locais, ignorando a distância física. Devido a isso, a quantidade de vagas oferecidas nesse modelo apresentou uma crescente; dados mostram que houve um crescimento de 136% no número de vagas disponíveis no ano de 2021 em relação à 2020 (GRANDCHAMP, 2022) e de 496% no primeiro semestre de 2022 em relação ao de 2021 (GOTO, 2022).

Por consequência, há uma maior facilidade para profissionais de diversas áreas encontrarem vagas de emprego remotas, incluindo os estágios supervisionados, como os realizados pelos alunos do Departamento de Ciência da Computação da Universidade Federal de Lavras (DCC/UFLA). O estágio supervisionado permite aos alunos colocar em prática conhecimentos teóricos adquiridos na universidade durante sua graduação, além de promover o desenvolvimento profissional e expandir seu conhecimento à respeito dos conteúdos passados em sala de aula.

Nesse contexto, o presente relatório de estágio descreve as atividades desenvolvidas durante o período do estágio supervisionado na empresa Embraer, na equipe de Engenharia de Software do setor da Qualidade. Essas atividades se relacionam às aprendidas em sala de aula como administração e manutenção de

bancos de dados, desenvolvimento de programas escaláveis com engenharia de software, controle de versões com *git*, manutenção e testes de software, entre outros. Por meio desse relatório, busca-se ainda destacar os benefícios dos conhecimentos teóricos juntamente à experiência prática adquiridos pelo acadêmico autor do presente relatório.

## 1.2 Sobre a Embraer

A Embraer é uma fabricante de aeronaves privada de capital aberto, fundada no ano de 1969, com aproximadamente 18 mil funcionários (SAMPAIO, 2020), faturamento de US\$ 4,5 bilhões no ano de 2022 (ARAUJO, 2023) com sede na cidade de São José dos Campos/SP.

Como missão, a Embraer procura gerar o maior valor possível para seus acionistas, garantindo a constante satisfação, através de integridade e qualidade, concentrando-se nas frentes da aviação comercial, aviação executiva e defesa. Como visão, a empresa procura se manter como referência global do mercado nacional e internacional, procurando liderança no mercado aeronáutico. A empresa tem como valores a “Ética e integridade estão em tudo que fazemos”, “Nossa gente é o que nos faz voar”, “Existimos para servir nossos clientes”, “Buscamos a excelência empresarial”, “Ousadia e inovação são a nossa marca”, “Atuação global é a nossa fronteira” e “Construímos um futuro sustentável”.

Sua estrutura organizacional procura manter altos padrões de transparência, eficácia e responsabilidade nos processos corporativos, tendo sua administração dividida em “Conselho de Administração e seus comitês de assessoramento”, “Conselho Fiscal” e “Diretoria”. O seu público alvo pode ser definido pelas companhias aéreas, empresas de aviação executiva, agências governamentais, operadores de aeronaves agrícolas e operadores de carga.

Em 5 de julho de 2018, a Embraer anunciou um acordo de intenções com a Boeing (outra fabricante de aeronaves) para a criação de uma *joint-venture*<sup>1</sup> e em 1 de janeiro de 2020, houve a conclusão do desmembramento de sua divisão de aeronaves comerciais para consumir este acordo. Segundo o vice-presidente executivo financeiro e de relações com investidores, Antônio Carlos Garcia, essa divisão foi feita ao aportar os passivos, ativos, bens, obrigações e direitos da unidade para uma nova empresa (VINHOLES, 2020).

Entretanto, em 24 de abril de 2020, o acordo de *joint-venture* foi rescindido unilateralmente pela Boeing (INFOMONEY, 2020). Dado o ocorrido, a Embraer deu início à um projeto interno para reintegração completa do negócio de aviação comercial (AGUIAR, 2022). Neste sentido, ferramentas de software internas, tais como *Cockpit* e *Weighing*, são necessárias para realizar o controle das diferentes etapas da fabricação de uma aeronave, bem como para comunicação do progresso com seus clientes.

O software *Cockpit* é composto por diversos módulos, tais como de transferência entre hangares, administração de inconsistências, manutenção de aeronaves, avaliação do serviço prestado, liberação de acesso para terceiros. Já o *Weighing*, por sua vez, é composto por módulos responsáveis pela pesagem das aeronaves.

### 1.3 Objetivos

A partir da realização dos serviços da empresa, surge a necessidade de realizar alterações nos produtos de software existentes. Sendo assim, o estágio supervisionado relatado neste documento teve como objetivo os apresentados a seguir:

---

<sup>1</sup> Modelo de parceria comercial entre duas ou mais empresas, reunindo seus recursos para realizar tarefas específicas (WIKIPEDIA CONTRIBUTORS, 2023)

*Cockpit*. Como exemplos de manutenção de funcionalidades já existentes desenvolvidas pelo autor deste relatório, estão: (i) junções de todas as tabelas da base de dados da qualidade; (ii) atualização da nomeação de arquivos para realizar junções de servidores; (iii) atualização da estrutura dos menus; entre outros.

*Weighing*. Como exemplos alterações nos bancos de dados desenvolvidas pelo autor deste relatório nesse software, estão: (i) junções de todas as tabelas da base de dados da qualidade; (ii) correções relativas à alterações manuais nas bases de dados; entre outros.

#### **1.4 Estrutura do trabalho**

A organização do trabalho está feita da seguinte forma: no Capítulo 2, explicações básicas e conceitos são apresentados sobre as ferramentas e técnicas empregadas ao longo do estágio; no Capítulo 3, as atividades desenvolvidas durante o período do estágio, assim como os resultados obtidos são apresentados; no Capítulo 4, considerações finais são apresentadas.

## 2 REFERENCIAL TEÓRICO

Tendo em vista que o estágio teve como base a manutenção de sistemas de software e de bases de dados, alguns conceitos precisam ser definidos. Desta forma, neste capítulo, são apresentados os principais conceitos sobre *Kanban Board*, *Scrum*, *ASP .NET*, *SQL*, *C#*, *Pentaho Kettle Spoon*, *Python*, *Scrapy* e *Jira*.

### 2.1 Kanban Board

No contexto do desenvolvimento ágil de *software*, apenas um quadro e cartões são plenamente suficientes para o desenvolvimento (RADIGAN, 2019). Esse método de visualização é conhecido como *Kanban*, sendo uma aplicação do sistema de produção criado pela Toyota, no final dos anos 40, aplicada no desenvolvimento de software. Conceitualmente, o objetivo do *Kanban* é permitir a visualização do fluxo geral de desenvolvimento do *software* e a atribuição de tarefas.

Existem dois tipos distintos de quadro *Kanban*, o físico, em que podem ser utilizadas notas adesivas em um quadro branco, e o digital, implementado como um *software*. O quadro *Kanban* é utilizado para guiar os desenvolvedores da equipe em quais tarefas devem ser realizadas primeiro, além de ajudar os donos do produto a avaliarem o desempenho de cada membro (conceito *Scrum* que será apresentado na Seção 2.2) e permitir uma visão geral sobre o produto desenvolvido. Na Figura 2.1, é exemplificado um quadro *Kanban* digital na ferramenta com opções de planos *Jira*<sup>1</sup>, a qual foi utilizada durante o estágio do presente autor.

Este quadro utiliza o método *Kanban* permitindo a gestão do fluxo de trabalho de um projeto. O *Kanban*, por ser um sistema flexível, permite aos usuários, definirem a quantidade de colunas necessárias para controlar o projeto de forma satisfatória. Nesse caso, foram definidas 4 (quatro) colunas: a primeira, *BACKLOG*, descreve as tarefas que serão futuramente desenvolvidas, as quais são necessárias

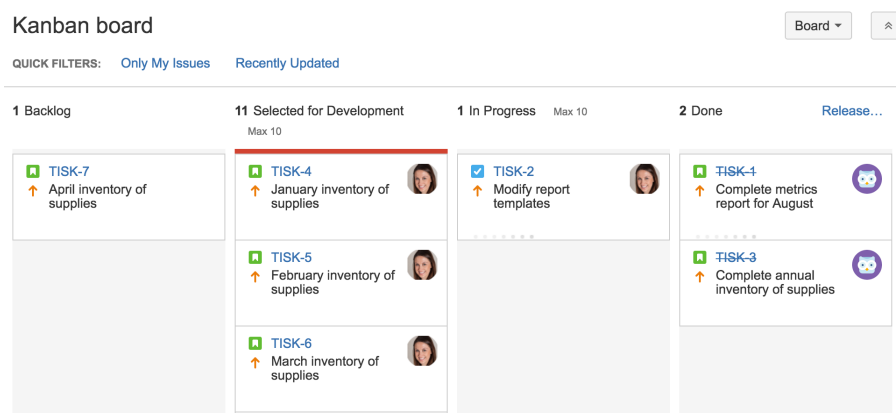
<sup>1</sup> Fonte: <https://www.atlassian.com/software/jira>



para completar o projeto e atingir o produto final. A segunda coluna, *SELECT FOR DEVELOPMENT*, representa as tarefas selecionadas do *Backlog* que devem ser desenvolvidas durante um determinado prazo; cada tarefa nessa coluna possui um responsável, mantendo assim, a rastreabilidade da mesma. A terceira coluna, *IN PROGRESS*, representa as tarefas que estão sendo trabalhadas, escolhidas pelos desenvolvedores a partir da coluna anterior. Por fim, a quarta coluna, *DONE*, representa as tarefas que já foram desenvolvidas, testadas e finalizadas.

Além disso, o *Kanban* pode utilizar duas metodologias de produção, sendo elas o *Kanban Puxado* e o *Kanban Empurrado*. No *Kanban Puxado*, as tarefas são realizadas conforme demanda do cliente. Conceitualmente, tem como base a demanda e estoque reduzido, procurando reduzir custos, eliminar desperdícios e estoque de produtos, assimilando-se à produção *Just in Time*<sup>2</sup>. Já no *Kanban Empurrado*, as tarefas são criadas com base em uma programação determinada anteriormente, desconsiderando a capacidade do time de desenvolvimento para adotá-las imediatamente. Tem como base maior planejamento e controle antecipado, alocação de recursos de forma mais precisa e simplicidade de implementação.

Figura 2.1 – Quadro kanban digital



Fonte: Maynard (2023).

<sup>2</sup> Modelo de gestão da produção que procura transportar, produzir e vender somente conforme necessário, sem manutenção de estoque.

No time de qualidade da Embraer, o *Kanban* era utilizado em conjunto com o *Scrum*, apresentado na Seção 2.2.

## 2.2 Scrum

Scrum é considerado um *framework*<sup>3</sup> de metodologia ágil para gestão de projetos de software (ORG, 2023). O Scrum é composto pelo “Time Scrum”, que por sua vez é dividido em três diferentes papéis, sendo eles *Product Owner*, *Scrum Master* e Desenvolvedores. O *Product Owner*, ou dono do produto, é responsável por garantir que o time está criando o produto mais fiel possível às expectativas das partes interessadas. O *Scrum Master* é responsável por garantir que o time seja o mais efetivo possível na utilização do Scrum. Os Desenvolvedores, por sua vez, são responsáveis por trabalhar no desenvolvimento do produto.

Esse “Time Scrum” participa de cinco eventos (ritos) distintos e, portanto, com diferente propósitos: *Sprint*, *Daily*, *Sprint Planning*, *Sprint Review* e *Sprint Retrospective*. A *Sprint* é considerada um ciclo de até um mês em que o trabalho é realizado. Dentro da *Sprint*, todos os outros eventos Scrum estão contidos e uma nova *Sprint* é iniciada após o término da outra. *Sprint Planning* é um evento realizado no início do ciclo da *Sprint*, com objetivo de planejar o trabalho que será desenvolvido durante este ciclo. *Daily* é uma reunião curta e diária, com objetivo de os desenvolvedores atualizarem o atual progresso em relação ao objetivo da *Sprint*, além de informarem possíveis impedimentos e realizar adaptações. *Sprint Review* é um evento que acontece no final da *Sprint*, em que o “Time Scrum” apresenta às partes interessadas o que foi atingido durante a *Sprint* e o que será feito na próxima. Por fim, *Sprint Retrospective* é uma reunião de refinamento em que o “Time Scrum” se reúne para conversar sobre a última *Sprint*, identificando mudanças que podem ser feitas para aumentar a eficácia do time.

---

<sup>3</sup> Conjunto de regras, crenças ou ideias que são utilizadas para lidar com problemas ou decidir o que fazer.

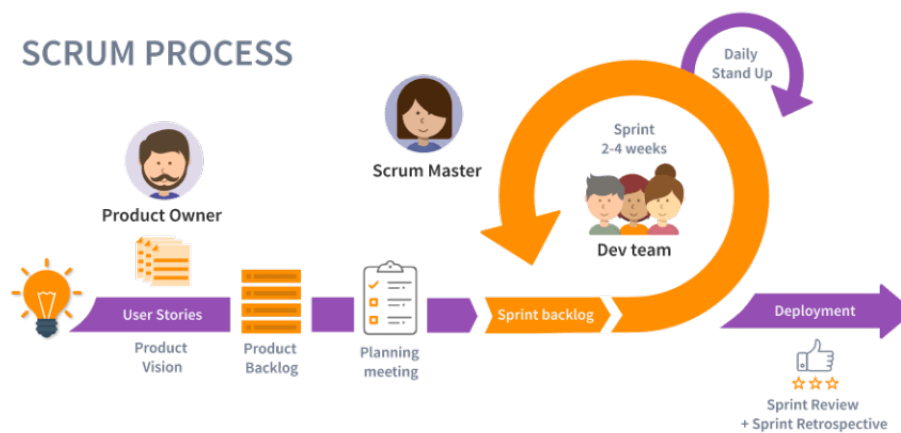
Dos artefatos Scrum, três são produzidos durante a *Sprint*, a saber: (i) *Product Backlog*, que é uma lista ordenada por prioridades, que define o que deve melhorar no produto; (ii) *Sprint Backlog*, que consiste de tarefas selecionadas do *Product Backlog* para serem desenvolvidas durante a atual *Sprint*; e (iii) os “Incrementos”, que são pequenas atividades que servem de caminho para atingir o objetivo do produto.

No “Time Scrum” da Embraer, os três papéis estavam presentes em todos os ritos. A *Sprint* possuía duas semanas de duração, em que todos os eventos estavam contidos. No início do primeiro dia da *Sprint*, ocorria a *Sprint Planning*, em que, utilizando a ferramenta Jira, tarefas eram selecionadas do *Backlog* para serem desenvolvidas durante a *Sprint* e os responsáveis por estas também eram atribuídos. Além disso, com exceção dos dias em que não haviam os eventos *Sprint Planning*, *Sprint Review* e *Sprint Retrospective*, uma *Daily* por dia era realizada, com duração de até 15 minutos. Na *Daily*, cada membro, em ordem alfabética, dizia o que foi feito desde a última *Daily*, o que será feito até a próxima e se há algum impedimento.

No último dia da *Sprint*, a *Sprint Review* ocorria, com duração de até uma hora. Nesta reunião, o time apresentava todas as tarefas desenvolvidas para o *Product Owner*, comparando o que foi proposto com o que realmente foi realizado. Ocorria também a coleta de *feedbacks* para serem aplicados na *Sprint* seguinte. Logo após a *Sprint Review*, a *Sprint Retrospective* acontecia, com duração média de até uma hora. Durante a *Sprint Retrospective*, era discutido o que deu certo durante a *Sprint*, tentando verificar os pontos que tornaram isso possível, e o que deu errado, buscando encontrar os motivos para isso ter acontecido e as ações que devem ser tomadas para mitigar os problemas. As ações que seriam tomadas para a próxima *Sprint* eram anotadas e adicionadas a um documento, facilitando a rastreabilidade e permitindo consultas futuras.

Quanto aos artefatos, o *Product Backlog* era bem volátil. A cada *Sprint*, novas tarefas apareciam, as quais deveriam ser priorizadas e transferidas para o *Backlog* da equipe. O *Sprint Backlog*, por sua vez, era selecionado do *Product Backlog* de forma priorizada, considerando a capacidade de trabalho da equipe, exemplificados na figura 2.2.

Figura 2.2 – Processo scrum



Fonte: Usemobile (2022).

## 2.3 Tecnologias utilizadas

Nesta seção, são apresentadas algumas das tecnologias utilizadas no desenvolvimento dos sistemas *Cockpit* e *Weighing*, durante o período de estágio na empresa *Embraer*.

### 2.3.1 CSS

A linguagem CSS (*Cascading Style Sheets*) é usada para descrever como os elementos de uma página web são exibidos na tela (CONTRIBUTORS, 2023). Para exemplificar, o resultado obtido na Figura 2.3, é devido a adição do código 2.4 na *tag head* da página principal de um programa básico *ASP.NET*, bem como a adição do atributo *class* de valor *estilizacao* na *tag h1*.

Figura 2.3 – Resultado da adição de CSS a um código base ASP.NET

## Página Simples ASP.NET

Uma página simples para demonstrar o funcionamento da tecnologia ASP.NET

Fonte: Do autor (2023).

A diferença de estilo em relação à adição do *CSS* é devida à adição das propriedades *background-color* e *color*, que alteram a cor do plano de fundo e da cor do título, respectivamente.

Figura 2.4 – Exemplo de código CSS

```
<style>
    .estilizacao {
        background-color: red;
        color: yellow;
    }
</style>
```

Fonte: Do autor (2023).

CSS foi utilizado no estágio para realizar a adição de estilização à novas páginas web do sistema *Cockpit*, buscando manter a identidade visual do sistema.

## 2.4 Banco de dados

Banco de dados é uma coleção de dados persistentes, usado pelos sistemas de uma determinada empresa (DATE; VIEIRA; LIFSCHITZ, 2004). Bancos de dados podem ser divididos, principalmente, em duas categorias: relacionais (baseados na linguagem SQL - *Structured Query Language*) e não relacionais (também conhecidos como *NoSQL*). Como o modelo de banco de dados utilizado no estágio foi o relacional, somente esse modelo será apresentado nesta seção.

Banco de dados relacional é um tipo específico de banco que permite o armazenamento e também garante o acesso a dados relacionados. São baseados no modelo relacional, isto é, uma forma de representar dados em tabelas, na qual cada linha da tabela é um registro com um identificador exclusivo, denominado chave primária. As colunas de uma tabela contêm atributos dos dados, que podem ser usados para estabelecer relações entre os duas ou mais tabelas (ORACLE, 2023).

Na Figura 2.5, é exemplificado o uso da linguagem *Oracle SQL* para realizar uma consulta em um banco de dados relacional fictício. Essa consulta retorna “id”, “nome” e “código de referência” dos fornecedores cujo nome da empresa inicia com “Empresa nacional de”.

Figura 2.5 – Exemplo de consulta Oracle SQL

```
SELECT
    id,
    nome,
    cod_ref
FROM
    fornecedores
WHERE
    UPPER(nome) LIKE 'EMPRESA NACIONAL DE%';
```

Fonte: Do autor (2023).

A linguagem *Oracle SQL* foi utilizada no estágio para realização das operações de leitura, inserção, atualização e remoção dos dados armazenados nos bancos de dados dos sistemas *Cockpit* e *Weighing*.

## 2.5 Python

*Python* é uma linguagem de programação que utiliza um interpretador ao invés de um compilador, possui orientação à objetos (OO), utiliza da tipagem dinâmica e é considerada de alto nível, suas estruturas de dados incorporadas de alto nível, combinadas com digitação dinâmica e ligação dinâmica, o tornam muito atraente para o desenvolvimento rápido de aplicativos (FOUNDATION, 2023).

O código da Figura 2.6 mostra a lógica para renomear arquivos de dentro de uma pasta.

Figura 2.6 – Exemplo de código Python para renomear arquivos

```
import os

PATH_TO_IMAGES = "C:/Images/"
EXTENSION = "png"

images = os.listdir (PATH_TO_IMAGES)

for i, image in enumerate(images):
    os.rename (f"{PATH_TO_IMAGES}/{image}",
               f"{PATH_TO_IMAGES}/{i}.{EXTENSION}")
```

Fonte: Do autor (2023).

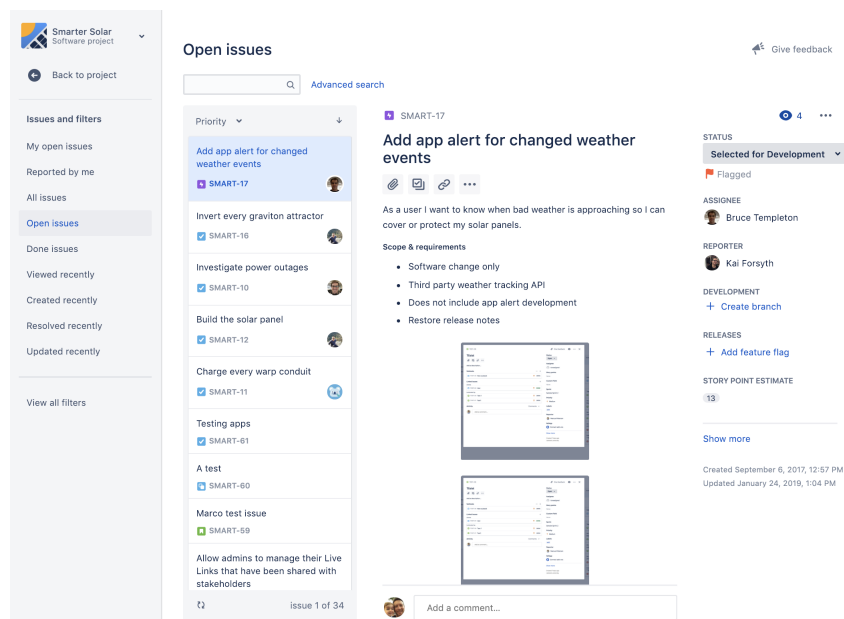
Na linha 1, tem-se a importação do módulo “os”, responsável por fornecer uma maneira simples de manipular arquivos. Após isso, definimos variáveis “PATH\_TO\_IMAGES” e “EXTENSION”, que serão utilizadas futuramente, sendo, o caminho dos arquivos no sistema operacional e sua extensão, respectivamente. Em seguida, é utilizada a função da biblioteca “os” chamada de “listdir”, que lista todos os arquivos de dentro de um caminho. Utilizamos essa lista, e iteramos por ela, utilizando “i” como índice e “image” como o nome da imagem. Sabendo disso, é utilizada a função da biblioteca “os” chamada de “rename”, a qual recebe dois parâmetros, o primeiro sendo o caminho e nome atual do arquivo e o segundo sendo o novo caminho e nome do arquivo. Com isso, utilizamos o caminho e nome atual de cada arquivo para o primeiro parâmetro e, para o segundo, definimos o mesmo caminho, mas agora com nome oriundo da variável “i”, que é um contador.

A linguagem *Python* foi utilizada durante o estágio para renomear arquivos que seriam transferidos de um servidor para o outro, a fim de resolver conflitos de nomeação de bases de dados distintas.

## 2.6 Gestão de Projetos com *Jira*

Para implementar as mudanças solicitadas por funcionários ou clientes da empresa Embraer, é preciso acessar o módulo de gestão de projetos da ferramenta *Jira* (Figura 2.7). Esse mesmo módulo é utilizado para gerenciamento de tarefas, encaminhamento de demanda para outros times, planejamento de trabalho e priorização.

Figura 2.7 – Seção de tarefas em aberto Jira



Fonte: Atlassian (2023).

Da forma como o “Time Scrum” funcionava na Embraer, o *Product Owner* selecionava as tarefas do módulo de gestão de projetos da ferramenta Jira e as priorizava, de acordo com a urgência e o esforço necessário para realizá-las. Logo após, o *Product Owner* adicionava essas tarefas ao *Kanban* da equipe. Após



a tarefa ser finalizada, um comentário deveria ser adicionado a ela e o Jira se encarregava de enviá-lo automaticamente ao criador da tarefa, juntamente com uma solicitação de aceite. Caso essa solicitação fosse aceita, a tarefa era então concluída, caso contrário, uma nova tarefa era automaticamente criada para resolução do problema.

### 3 ATIVIDADES REALIZADAS

Neste capítulo, são apresentadas algumas das atividades realizadas durante o período de estágio. Grande parte do tempo despendido foi para o sistema *Cockpit*, dada a sua importância, tamanho e complexidade. Além disso, algumas atividades também foram realizadas no sistema *Weighing*.

#### 3.1 Descrição das atividades admissionais

Nesta seção, são apresentadas as atividades realizadas para entrar na *Embraer* e o processo de integração.

Todo o processo de seleção foi realizado através da plataforma *Gupy*. A cultura da empresa valoriza, na seleção dos candidatos, a motivação para aprender e alinhamento cultural com os valores, por esse motivo, não há etapa de exame prático de programação. O recrutamento consistiu de 4 (quatro) etapas, e a aprovação de cada etapa liberava a seguinte, sendo estas:

- Cadastro: utilizada para realizar o cadastro inicial, com dados como nome completo, email, currículo, experiência, curso superior entre outras informações
- Teste de Inglês: utilizada para testar o nível de inglês do candidato, feito através de uma prova online de múltipla escolha
- Carta de Motivação: utilizada para o candidato escrever suas motivações e por qual propósito desejava a vaga, o motivo para ser aceito
- Dinâmica Online: dinâmica online onde diversos candidatos participavam e perguntas eram realizadas a cada um deles, procurando entender o nível de experiência e alinhamento do candidato com a vaga

Após passar por todas as etapas e receber o convite para participar do programa de estágio, o candidato deveria providenciar toda a documentação necessá-

ria para a admissão, com dados pessoais e documentos acadêmicos comprovando o ingresso em curso superior. O processo de acolhimento foi realizado durante as primeiras semanas do estágio, em que o time responsável pela integração dos estagiários realizava apresentações explicando o funcionamento do programa de estágio e encaminhava os estagiários para acessar o portal de aprendizado interno. Esse portal apresentava inúmeros treinamentos, em formato de videoaulas, que todos os funcionários deveriam realizar para compreender a estrutura interna. Além disso, o autor desse relatório recebeu temporariamente, para realização das atividades, o notebook *Lenovo ThinkPad*, que permitiu a realização das tarefas sem impedimentos de *hardware*.

A alocação de projetos era definida pelo gestor da equipe, levando em conta os conhecimentos do estagiário bem como necessidade de capacidade em certos projetos. A equipe focada em Engenharia de Software, pertencentes ao time da “Qualidade”, ao qual o presente autor participou, funcionava como um único time em que as tarefas eram divididas e compartilhadas.

### **3.2 Descrição das atividades desenvolvidas no *Cockpit***

Nesta seção, são apresentadas as atividades realizadas no sistema *Cockpit*, bem como as dificuldades enfrentadas e conhecimentos adquiridos. Esse sistema permite cadastrar, consultar, atualizar e remover dados relacionados à produção, qualidade e entrega de diversas aeronaves da companhia *Embraer*.

#### **3.2.1 Junções das tabelas da base de dados da qualidade**

Por conta da rescisão do acordo de *joint-venture* com a *Boeing* (conforme descrito na Seção 1.2), a *Embraer* optou por reintegrar a divisão de aeronaves comerciais. O objetivo da reintegração era a recuperação de sinergia da empresa, garantindo maiores benefícios operacionais, eliminando, também, ineficiências fiscais através de uma gestão integrada, menos complexa e mais ágil (NOTÍCIAS,

2022). O *Cockpit*, responsável pela administração de processos de construção da aeronave, revisão direta com os clientes e geração de relatórios, possuía uma única base de dados e, ao realizar a divisão da *Embraer* em duas empresas, esta base foi duplicada para a divisão de aeronaves comerciais (*Cockpit Commercial*). Com o objetivo da reintegração, se tornou necessário a junção das tabelas da base de dados da qualidade. Conforme, a *Embraer* foi a única responsável por arcar com os custos da reintegração.

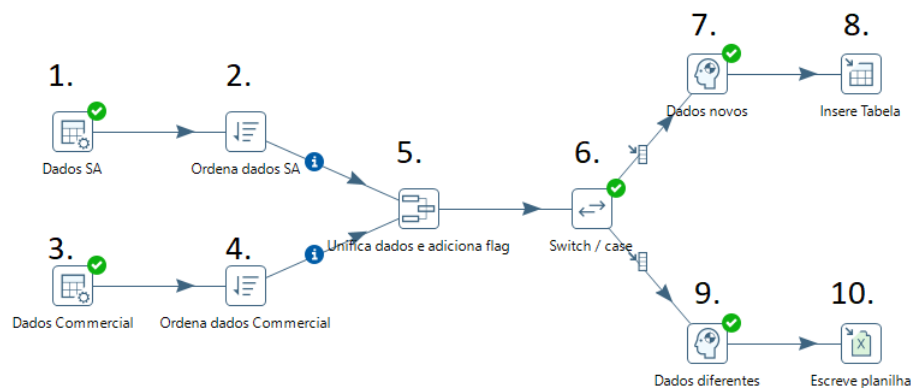
A Figura 3.1 apresenta um fluxo, criado pelo autor deste relatório, para análise dos dados das diferentes bases. Os passos 1 e 3 são executados em paralelo, e são responsáveis por extrair os registros da base de dados da *SA* e da *Commercial*, respectivamente. Os passos 2 e 4 também são executados em paralelo, ordenando os dados de cada base, em relação à sua chave primária. O passo 5 é executado, unindo os registros de ambas as bases e dividindo-os em duas categorias, a saber:

- Dados novos: dados que existem na base *Cockpit Commercial* e não na base *Cockpit SA*, sem conflito de chave primária
- Dados diferentes: dados que existem na base *Cockpit Commercial* e não na base *Cockpit SA*, com conflito de chave primária

Os dados novos eram inseridos diretamente na base *Cockpit SA*, conforme verificado nos passos 7 e 8. Os dados diferentes eram adicionados a uma planilha (passo 10), que foi utilizada para definir regras de remoção de conflito.

A escolha da utilização de planilhas como resultado do fluxo, foi por se tratar de uma ferramenta usada vastamente na empresa. Com base na planilha gerada do passo 10 da Figura 3.1, uma nova planilha “DE-PARA” era definida (Figura 3.2). O objetivo principal da planilha era, utilizar dos registros diferentes e com mesma chave primária, e criar uma nova chave primária, com objetivo de permitir a inserção no banco de dados do *Cockpit SA*, adicionando um desvio. O exemplo da Figura 3.2 apresenta as chaves primárias que haviam conflito entre

Figura 3.1 – Fluxo para analisar e unir bases de dados distintas



Fonte: Do autor (2023).

as duas tabelas, representados pela coluna “ANTIGA\_CHAVE”. Por sua vez, a coluna “NOVA\_CHAVE” apresenta o novo valor que a chave deveria tomar, ao registro ser inserido no banco de dados do *Cockpit SA*.

Após a finalização da análise das tabelas com dados diferentes, diversas consultas *SQL* eram realizadas para substituir as chaves antigas pelas novas ao inserir os dados na base *Cockpit SA*.

As consultas das figuras 3.3 e 3.4 apresentam a sintaxe para a criação de quatro tabelas “foods1”, “dishes1”, “foods2” e “dishes2”. Os dados inseridos e as tabelas criadas são simbólicas para exemplificar o processo de unificação das bases de dados realizado na *Embraer*. A tabela “dishes1”, possui referência para a tabela “foods1” através da chave estrangeira chamada de “FoodID”. Concomitantemente, a tabela “dishes2”, possui referência para a tabela “foods2” através da chave estrangeira chamada de “FoodID”.

Como exemplo das uniões das bases de dados realizadas durante o estágio na *Embraer*, as tabelas criadas nas figuras 3.3 e 3.4 serão usadas realizar o mesmo processo.

Figura 3.2 – Planilha “DE-PARA”

ANTIGA_CHAVE	NOVA_CHAVE
1	1001
2	1002
3	1003
4	1004
5	1005
6	1006
7	1007
8	1008
9	1009
10	1010
11	1011
12	1012
13	1013
14	1014
15	1015
16	1016
17	1017
18	1018
19	1019

Fonte: Do autor (2023).

As consultas da figura 3.5 adicionam uma nova coluna à tabela “foods1” com valor nulo para cada registro.

A consulta da figura 3.6 identifica e insere na tabela “foods1” os registros da tabela “foods2” que não existem em “foods1”. Para isso, realiza o comando de inserção “INSERT INTO” para adicionar registros com os dados “ID”, “Food” e “OLD\_ID”. Filtrando esses dados através da operação “LEFT JOIN”, utilizando como parâmetro o “ID” de cada registro. Essa operação seleciona somente os registros de “foods2” que não existem em “foods1”

A consulta da figura 3.7 identifica e insere na tabela “foods1” os registros da tabela “foods2” que possuem conflitos em “foods1”. A princípio, uma variável de nome “@MaxID” é criada temporariamente para a consulta *SQL*. À essa

Figura 3.3 – Consultas SQL para criação de tabelas de exemplo *foods*

```

CREATE TABLE foods1 (
    ID INT PRIMARY KEY,
    Food NVARCHAR(255)
);

INSERT INTO foods1 (ID, Food)
VALUES
    (1, 'Pizza'),
    (2, 'Hamburger'),
    (3, 'Macarrao'),
    (4, 'Salada'),
    (5, 'Sushi');

CREATE TABLE foods2 (
    ID INT PRIMARY KEY,
    Food NVARCHAR(255)
);

INSERT INTO foods2 (ID, Food)
VALUES
    (1, 'Pizza'),
    (4, 'Brocolis'),
    (5, 'Macaron'),
    (6, 'Frango');

```

Fonte: Do autor (2023).

variável, é atribuído o maior valor do “ID” da tabela “foods1”. Após instanciar a variável, há o comando de inserção “INSERT INTO” para adicionar registros com os dados “ID”, “Food” e “OLD\_ID”. Em sequência, a instrução “SELECT” é realizada, em que, gera um novo “ID” para a inserção, utilizando da variável criada anteriormente “MaxID” e somando com o número gerado da coluna, garantindo que o “ID” seja um valor único. A instrução “FROM”, especifica os registros da tabela “foods2” que devem ser inseridas em “foods1”, os quais, nesse caso, serão os registros com mesmo “ID” nas duas tabelas. Para não inserir registros idênticos em relação à coluna “Food”, a instrução “WHERE” é executada, delimitando somente aos registros em que o valor de “foods” da tabela “foods1” é diferente da tabela “foods2”.

Figura 3.4 – Consultas SQL para criação de tabelas de exemplo *dishes*

```

CREATE TABLE dishes1 (
    DishID INT PRIMARY KEY,
    DishName NVARCHAR(255),
    FoodID INT,
    FOREIGN KEY (FoodID) REFERENCES foods1(ID)
);

INSERT INTO dishes1 (DishID, DishName, FoodID)
VALUES
    (1, 'Pizza Margherita', 1),
    (2, 'Cheese Burguer', 2),
    (3, 'Macarrao Carbonara', 3),
    (4, 'Salada Grega', 4),
    (5, 'Combinado de sushi', 5);

CREATE TABLE dishes2 (
    DishID INT PRIMARY KEY,
    DishName NVARCHAR(255),
    FoodID INT,
    FOREIGN KEY (FoodID) REFERENCES foods2(ID)
);

INSERT INTO dishes2 (DishID, DishName, FoodID)
VALUES
    (1, 'Pizza Margherita', 1),
    (4, 'Salada de Brocolis', 4),
    (5, 'Bolo de Macaron', 5),
    (6, 'Frango Frito', 6);

```

Fonte: Do autor (2023).

Figura 3.5 – Adiciona nova coluna OLD\_ID na tabela *foods1*

```

ALTER TABLE foods1
ADD OLD_ID INT NULL;

```

Fonte: Do autor (2023).

As consultas da figura 3.5 adicionam uma nova coluna à tabela “*dishes1*” com valor nulo para cada registro.

A consulta da figura 3.9, similar à 3.6 identifica e insere na tabela “*dishes1*” os registros da tabela “*dishes2*” que não existem em “*dishes1*”. Para isso, realiza o comando de inserção “*INSERT INTO*” para adicionar registros com os dados



Figura 3.6 – Adiciona os registros na tabela foods1 sem conflito

```
INSERT INTO foods1 (ID, Food, OLD_ID)
SELECT f2.ID, f2.Food, f2.ID
FROM foods2 AS f2
LEFT JOIN foods1 AS f1 ON f2.ID = f1.ID
WHERE f1.ID IS NULL;
```

Fonte: Do autor (2023).

Figura 3.7 – Adiciona os registros na tabela foods1 com conflito

```
DECLARE @MaxID INT;
SELECT @MaxID = MAX(ID) FROM foods1;

INSERT INTO foods1 (ID, Food, OLD_ID)
SELECT @MaxID + ROW_NUMBER()
      OVER (ORDER BY f2.ID), f2.Food, f2.ID
FROM foods1 AS f1
INNER JOIN foods2 AS f2 ON f1.ID = f2.ID
WHERE f1.Food <> f2.Food;
```

Fonte: Do autor (2023).

Figura 3.8 – Adiciona nova coluna FROM\_DISHES2 na tabela dishes1

```
ALTER TABLE dishes1
ADD FROM_DISHES2 BIT NULL;
```

Fonte: Do autor (2023).

Figura 3.9 – Adiciona os registros na tabela dishes1 sem conflito

```
INSERT INTO dishes1 (DishID, DishName, FoodID, FROM_DISHES2)
SELECT d2.DishID, d2.DishName, d2.FoodID, 1
FROM dishes2 AS d2
LEFT JOIN dishes1 AS d1 ON d2.DishID = d1.DishID
WHERE d1.DishID IS NULL;
```

Fonte: Do autor (2023).

“ID”, “Food” e “FROM\_DISHES2”, deixando por padrão, o valor de “FROM\_DISHES2” como “1”. Filtrando esses dados através da operação “LEFT JOIN”, utilizando como parâmetro o “ID” de cada registro. Essa operação seleciona somente os registros de “dishes2” que não existem em “dishes1”.

Figura 3.10 – Adiciona os registros na tabela dishes1 com conflito

```

DECLARE @MaxID2 INT;
SELECT @MaxID2 = MAX(DishID) FROM dishes1;

INSERT INTO dishes1
    (DishID, DishName, FoodID, FROM_DISHES2)
SELECT @MaxID2 + ROW_NUMBER() OVER
    (ORDER BY d2.DishID), d2.DishName, d2.FoodID, 1
FROM dishes1 AS d1
INNER JOIN dishes2 AS d2 ON d1.DishID = d2.DishID
WHERE d1.DishName <> d2.DishName;

```

Fonte: Do autor (2023).

A consulta da figura 3.10 identifica e insere na tabela “dishes1” os registros da tabela “dishes2” que possuem conflitos em “dishes1”. A princípio, uma variável de nome “@MaxID2” é criada temporariamente para a consulta *SQL*. À essa variável, é atribuído o maior valor do “ID” da tabela “dishes1”. Após instanciar a variável, há o comando de inserção “INSERT INTO” para adicionar registros com os dados “ID”, “Food” e “FROM\_DISHES2”. Em sequência, a instrução “SELECT” é realizada, em que, gera um novo “ID” para a inserção, utilizando da variável criada anteriormente “MaxID2” e somando com o número gerado da coluna, garantindo que o “ID” seja um valor único. A instrução “FROM”, especifica os registros da tabela “dishes2” que devem ser inseridas em “dishes1”. Que nesse caso, serão os registros com mesmo “ID” nas duas tabelas. Para não inserir registros idênticos em relação à coluna “DishName”, a instrução “WHERE” é executada, delimitando somente aos registros em que o valor de “DishName” da tabela “dishes1” é diferente da tabela “dishes2”.

Figura 3.11 – Atualiza a referência da chave estrangeira

```

UPDATE dishes1
SET FoodID = f1.ID
FROM foods1 as f1
WHERE FoodID = f1.OLD_ID AND FROM_DISHES2 = 1;

```

Fonte: Do autor (2023).

A consulta da figura 3.11 atualiza a chave estrangeira dos registros da tabela “dishes1” que foram inseridos na consulta 3.10, para referenciar a tabela “foods1” corretamente.

Figura 3.12 – Remove colunas desnecessárias

```
ALTER TABLE foods1 DROP COLUMN old_id;  
ALTER TABLE dishes1 DROP COLUMN FROM_DISHES2;
```

Fonte: Do autor (2023).

Por fim, as consultas da figura 3.12, removem as colunas desnecessárias “OLD\_ID” da tabela “foods1” e “FROM\_DISHES2” da tabela “dishes1” que foram adicionadas para facilitar a união das tabelas.

O esforço para o desenvolvimento da tarefa foi considerado alto, por se tratar de uma grande quantidade de tabelas com inúmeros registros cada. Para cada tabela, os fluxos para análise e união das bases de dados eram copiados, e cada passo era individualmente modificado. O trabalho realizado através do *SQL*, em suma, foi o de realizar a junção de bases de dados previamente separadas, com objetivo de realizar a integração, facilitar a visualização dos dados e reduzir custos operacionais. Após a execução, para cada tabela, do fluxo de análise e união das bases de dados, as informações eram validadas. Essa validação, consistia da consulta das bases de dados através de comandos *SQL*, para a verificação se os dados foram inseridos nas bases de dados, conforme exemplificado na Figura 3.13, onde, todas as chaves, com valor acima de “1000” eram selecionados. Além disso, a validação também envolvia o acesso da plataforma *Cockpit SA*, verificando se as tabelas exibidas no *website* estavam exibindo os mesmos registros e mesma quantidade da base de dados. Também, eram realizadas videochamadas com usuários da plataforma, solicitando a confirmação de que os dados inseridos no *website* estavam congruentes.

### 3.2.2 Atualização da nomeação de arquivos para realizar junções de servidores

Figura 3.13 – Consulta SQL para validação dos registros

```
SELECT
    *
FROM
    TABELA_EMBRAER_1
WHERE
    CHAVE > 1000
```

Fonte: Do autor (2023).

Uma das dificuldades encontradas durante o estágio foi a necessidade de atualizar a nomeação dos arquivos de texto, imagens e PDFs para realizar junções de diferentes tabelas dos bancos de dados.

Por conta de o sistema *Cockpit* possuir diversos arquivos (arquivos de texto, imagens e PDFs) em seus registros, estes, eram salvos em servidores e permaneciam acessíveis para os usuários através da ferramenta. Entretanto, por conta da rescisão do acordo de *joint-venture* (conforme descrito na seção 1.1) e junção das bases, os arquivos, que estavam em diferentes bases para cada empresa (SA e Commercial) deveriam ser unificados.

Para isso, considerando arquivos com mesmo nome e em diferentes servidores, as regras estabelecidas para a junção das tabelas na base de dados da qualidade foram adaptadas para renomear os arquivos, permitindo, assim, a sua transferência de forma organizada e sem conflitos entre as bases de dados.

Por se tratar de milhões de arquivos, não havia como realizar a migração de forma manual. Portanto, para esta tarefa, utilizou-se a linguagem Python para realizar a renomeação e transferência dos arquivos.

No código 3.14, arquivos são copiados do servidor *Commercial* para o servidor *SA*, renomeando com base em um dicionário do *Python*, onde os arquivos com nomes duplicados, eram copiados para a nova base, porém, com os novos

Figura 3.14 – Código python para renomear arquivos e mover entre diferentes locais

```
import shutil

BASE_SA_PATH = "E:/Quality/files"
BASE_CO_PATH = "F:/Quality/files"

NAMES_CHANGE = {
    "oldName0.png": "newName0.png",
    "oldName1.png": "newName1.png",
    "oldName2.png": "newName2.png"
}

for fileToRename in NAMES_CHANGE.keys():
    shutil.copy(f"{BASE_CO_PATH}/{fileToRename}",
               f"{BASE_SA_PATH}/{NAMES_CHANGE[fileToRename]}")
```

Fonte: Do autor (2023).

nomes. Para isso, a função “copy” do módulo “shutil” foi utilizada, simplificando a tarefa de renomear e copiar os arquivos em uma única linha de código.

O desenvolvimento dessa tarefa não foi muito custosa, por se tratar de uma atividade simples e que necessitava de conhecimento apenas na linguagem de programação *Python*. A escolha da tecnologia se deu pela facilidade de manipulação de arquivos e confiabilidade.

### 3.2.3 Atualização da estrutura dos menus

Durante a junção das bases de dados do sistema *Cockpit*, foi observado que outras seções do sistema não estavam sendo acessadas ou não possuíam utilidade prática. Essas seções, referem-se a páginas do software que os usuários não acessavam. Essa observação foi relevante para o processo de melhoria contínua da ferramenta.

Por esse motivo, um processo minucioso de validação foi realizado, avaliando cada seção do sistema e seu uso pelos usuários. Essa validação, consistiu em analisar o registro do banco de dados verificando o acesso de cada página pelos usuários. Após a validação, foi constatado que haviam seções sem registros

de atividade de usuários, conseqüentemente, essas foram removidas. Entretanto, a remoção, não foi feita de forma dinâmica, a solução aplicada foi a de remover o acesso dos usuários a essas seções, fazendo também, com que não aparecessem no menu de navegação.

### **3.3 Descrição das atividades desenvolvidas no *Weighing***

Nesta seção, são apresentadas as atividades realizadas nas páginas de responsabilidade da qualidade do sistema *Weighing*, bem como as dificuldades enfrentadas e conhecimentos adquiridos. Esse sistema era utilizado pelos times de segurança e de montagem de aeronaves, o qual permite cadastrar, consultar, atualizar e remover de dados relacionados à pesagem das aeronaves da *Embraer*. Esses dados eram utilizados para verificar se as aeronaves estavam nas conformidades estabelecidas pela organização, em especial, se o peso correspondia aos parâmetros esperados, permitindo avaliar se algum ajuste futuro deveria ser realizado.

#### **3.3.1 Junções das tabelas da base de dados da qualidade**

Para realização da junção das tabelas da base de dados da qualidade, a mesma estratégia da seção 3.2.1 foi aplicada, alterando somente as bases de dados utilizadas. Por se tratar de uma estratégia já implementada e por ser uma base de dados menor, a junção das tabelas foi pouco custosa.

#### **3.3.2 Correções relativas à alterações manuais nas bases de dados**

Após a união das bases de dados do sistema *Weighing*, foi verificado que alguns arquivos (arquivos de texto, imagens e PDFs) anteriormente adicionados não estavam acessíveis pela interface.

Diferente do problema enfrentado na seção 3.2.2, foi identificado que a ausência dos arquivos estava diretamente relacionada à inserções manuais efetua-

das diretamente nos bancos de dados, em vez de realizar as inserções por meio do sistema web.

Para resolver esse problema, o servidor foi restaurado com os dados antigos e esses arquivos foram novamente adicionados às bases de dados. Para fazer isso, lógica similar à aplicada na Seção 3.2.2 foi utilizada. Conforme visto no código 3.15, a mesma lógica da Seção 3.2.2 parece implementada, porém, há uma validação para verificar se o arquivo existe ou não no servidor antes de copiá-lo.

Figura 3.15 – Código python para copiar arquivos de uma pasta para outra ignorando cópias

```
import shutil
import os

RESTORED_PATH = "E:/Restored"
BASE_PATH = "E:/Files"

BASE_FILES = os.listdir(BASE_PATH)
RESTORED_FILES = os.listdir(RESTORED_PATH)

for fileName in RESTORED_FILES:
    if fileName not in BASE_FILES:
        shutil.copy(f"{RESTORED_PATH}/{fileName}",
                    f"{BASE_PATH}/{fileName}")
```

Fonte: Do autor (2023).

Por se tratar de uma estratégia já implementada e simples, a tarefa demandou pouco esforço e foi completada rapidamente.

## 4 CONCLUSÃO

Este trabalho apresentou as principais atividades realizadas pelo estagiário na empresa *Embraer*, no período de abril de 2021 à abril de 2022. O objetivo principal foi a junção de duas bases de dados, identificando padrões, resolvendo discrepâncias e aplicando estratégias para realizar a unificação destas, também, realizando manutenção das aplicações já existentes.

Ter a experiência de poder trabalhar em uma empresa real é algo extremamente enriquecedor, que permite crescimento e amadurecimento profissional e pessoal. Ademais, permite aos estudantes aprender a lidar com problemas reais e como resolvê-los de forma ágil mantendo a qualidade. Outro fator relevante é permitir colocar em prática habilidades aprendidas durante o curso superior.

As principais dificuldades encontradas foram referentes à utilização de novas linguagens de programação como *C#*, ferramentas como *Visual Studio*, e aprimoramento de conhecimentos em administração em bases de dados. Esses desafios foram superados com treinamentos e ensinamentos passados pelo time de Engenharia de Software da Qualidade, bem como estudos autônomos através de websites como *Udemy*. Nesse período, a manutenção de bases de dados foi um tarefa que perdurou durante todo o estágio, permitindo adquirir conhecimentos sólidos em linguagens *SQL*.

Verifica-se também que diversas disciplinas são relevantes para a atuação nas atividades do estágio, podendo destacar: Introdução aos Algoritmos, Estruturas de Dados, Modelagem e Implementação de Software, Introdução a Sistemas de Banco de Dados, Práticas de Programação Orientada a Objetos, Engenharia de Software, Interação Humano-Computador. Tais disciplinas garantem ao aluno conhecimento para atuar em diversas áreas do desenvolvimento, permitindo, aprimoramento de sua lógica, noções de ferramentas e suas utilidades, aprendizado de boas práticas e estruturação de software e noções em aplicações de bancos de dados para armazenamento de informações.



Após finalização do estágio, é possível constatar o impacto positivo que o trabalho em equipe permite ao estagiário, garantindo desenvolvimento dessa habilidade essencial para o mercado de trabalho, bem como, acelerando o desenvolvimento dos formandos. Em suma, participar de um programa de estágio, pode garantir ao estudante diversos benefícios em relação ao seu preparo para o mercado de trabalho, garantindo aplicação de conhecimentos teóricos aprendidos durante o curso superior.

## REFERÊNCIAS

AGUIAR, V. Embraer (embr3) reintegra o setor de aviação comercial após o fiasco com a boeing. e agora, como ficam as ações? **SeuDinheiro**, São Paulo, 26 jan. 2022. Bolsa dólar. Disponível em: <<https://www.seudinheiro.com/2022/bolsa-dolar/embraer-embr3-reintegra-aviacao-comercial-apos-boeing/>>. Acesso em: 02 out. 2023.

ARAÚJO, G. Embraer espera receita líquida maior em 2023 com aumento de entregas. **CNN Brasil**, São Paulo, 10 mar. 2023. Economia. Disponível em: <<https://www.cnnbrasil.com.br/economia/embraer-espera-receita-liquida-maior-em-2023-com-aumento-de-entregas/>>. Acesso em: 02 out. 2023.

ATLASSIAN. **View a project's issues**. Sydney, 2023. Disponível em: <<https://support.atlassian.com/jira-work-management/docs/view-a-projects-issues>>. Acesso em: 02 out. 2023.

CONTRIBUTORS, M. **Cascading style sheets**. San Francisco, 2023. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/CSS>>. Acesso em: 02 out. 2023.

DATE, C. J.; VIEIRA, A.; LIFSCHITZ, S. **Introdução a sistemas de bancos de dados**. 8. ed. Rio de Janeiro: Campus, 2004.

FOUNDATION, P. S. **What is python? Executive summary**. Beaverton, 2023. Disponível em: <<https://www.python.org/doc/essays/blurb/>>. Acesso em: 02 out. 2023.

GOTO, M. Vagas para home office cresceram 496% em 2022, mostra pesquisa: Os cargos que tiveram a maior adesão ao trabalho remoto foram os de programador, consultor comercial e analista de testes, segundo estudo da catho. **Época Negócios**, São Paulo, 09 ago. 2022. Carreira. Disponível em: <<https://epocanegocios.globo.com/Carreira/noticia/2022/08/vagas-para-home-office-cresceram-496-em-2022-mostra-pesquisa.html>>. Acesso em: 02 out. 2023.

GRANDCHAMP, L. Vagas de emprego na área de tecnologia aumentaram 136% em 2021. **Rede Jornal Contábil**, Araguari, 05 mai. 2022. Carreira. Disponível em: <<https://www.jornalcontabil.com.br/vagas-de-emprego-na-area-de-tecnologia-aumentaram-136-em-2021/>>.

INFOMONEY, E. Boeing desiste de acordo de joint-venture com embraer: A boeing e a embraer irão manter o contrato vigente relativo à comercialização e manutenção conjunta da aeronave militar c-390 millenium. **InfoMoney**, São Paulo, 25 abr. 2020. Mercados. Disponível em: <<https://www.infomoney.com>>.

br/mercados/boeing-desiste-de-acordo-de-joint-venture-com-embraer/>. Acesso em: 02 out. 2023.

MAYNARD, C. **Learn kanban with jira software**. Sydney, 2023. Disponível em: <<https://www.atlassian.com/agile/tutorials/how-to-do-kanban-with-jira-software>>. Acesso em: 02 out. 2023.

NOTÍCIAS, E. C. A embraer conclui com sucesso reintegração do negócio de aviação comercial. **Embraer**, São José dos Campos, 26 jan. 2022. Corporativo. Disponível em: <<https://embraer.com/br/pt/noticias?slug=1206980-embraer-conclui-com-sucesso-reintegracao-do-negocio-de-aviacao-comercial>>. Acesso em: 02 out. 2023.

ORACLE. **O que é um banco de dados relacional (RDBMS)?** Austin, 2023. Disponível em: <<https://www.oracle.com/br/database/what-is-a-relational-database/>>. Acesso em: 02 out. 2023.

ORG, S. **What is scrum?** Burlington, 2023. Disponível em: <<https://www.scrum.org/resources/what-is-scrum>>. Acesso em: 02 out. 2023.

RADIGAN, D. **What is kanban?** Sydney, 2019. Disponível em: <<https://www.atlassian.com/agile/kanban>>. Acesso em: 25 out. 2023.

SAMPAIO, L. Boeing x embraer: após rompimento bilionário, americana avança sobre talentos da brasileira. **InfoMoney**, São Paulo, 25 abr. 2020. Mercados. Disponível em: <<https://www.infomoney.com.br/carreira/boeing-x-embraer-apos-rompimento-bilionario-americana-avanca-sobre-talentos-da-brasileira/>>. Acesso em: 13 mar. 2023.

USEMOBILE. **Metodologia scrum para desenvolvimento de aplicativos**. Ouro Preto, 2022. Disponível em: <<https://usemobile.com.br/metodologia-scrum-desenvolvimento/>>. Acesso em: 02 out. 2023.

VINHOLE, T. Embraer conclui separação da divisão de aviação comercial. **Airway**, São Paulo, 02 jan. 2020. Aviação comercial. Disponível em: <<https://www.airway.com.br/embraer-conclui-separacao-da-divisao-de-aviacao-comercial/>>. Acesso em: 02 out. 2023.

WIKIPEDIA CONTRIBUTORS. **Embraer**. San Francisco, 2023. Disponível em: <<https://en.wikipedia.org/w/index.php?title=Embraer&oldid=1112358297>>. Acesso em: 02 out. 2023.

XTRATEGIE FIA, C. D. T. **Pesquisa gestão de pessoas na crise COVID-19: relatório final**. Uberlândia, 2023. Disponível em: <<https://siteatmosfera.xtrategie.com.br/wp-content/uploads/2021/06/Relatorio-Final-Pesquisa-COVID19-compacto.pdf>>. Acesso em: 02 out. 2023.