

LUCAS SANDRO ROTERMEL FRANCO

PROJETOS PARA AUXÍLIO EM ENTREGA DO PRODUTO AO CLIENTE

Relatório técnico de estágio/emprego apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Câmpus Gaspar do Instituto Federal de Santa Catarina para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Leonardo Ronald Perin Rauta

Supervisor: Johnny Régis Fusinato

Gaspar

2022

RESUMO

O documento a seguir apresenta informações sobre o período de acompanhamento das atividades realizadas pelo aluno em seu emprego. São abordados junto com o desenvolvimento do projeto o tema *DevOps* apresentando algumas das ferramentas utilizadas, a empresa onde o aluno trabalhou e análises sobre a graduação que o aluno se encontra matriculado. Os projetos abordados são principalmente relacionados com a atualização do sistema em clientes, de forma ordenada por execução, os projetos falam sobre: criação de *endpoints* utilizados para validar versões de aplicações no cliente, análise da criação de *sequences* em SQL ao atualizar o sistema e um projeto referente aos dados que o cliente recebe sobre o *build* que o cliente possui instalado. Todos os projetos são documentados expondo o processo realizado desde a concepção até sua entrega, apresentando validações criadas para que cada projeto tivesse sua entrega mensurada com sucesso ao final e também apresentar seus pontos mais importantes tanto na modelagem quanto no desenvolvimento.

Palavras-Chave: Entregas. Atualização. Performance.

LISTA DE FIGURAS

Índice de figuras

Figura 1: Diagrama de Atividades.....	9
Figura 2: Exemplo de uma sequence criada com o NoCache.....	10
Figura 3: Exemplo de uma sequence criada com cache informado.....	11

SUMÁRIO

Índice

1 Introdução.....	4
2 Empresa onde o estágio/emprego foi realizado.....	4
3 Setor da empresa onde o estágio/emprego foi realizado.....	5
4 Período em que o estágio/emprego foi realizado.....	5
5 Métodos, Técnicas e Tecnologias Utilizadas.....	5
5.1 Métodos.....	5
5.2 Técnicas.....	6
5.3 Tecnologias.....	6
6 Projetos em que o aluno atuou durante o semestre.....	7
7 Descrição das Atividades Realizadas.....	7
7.1 Criação de <i>Endpoint</i> para compatibilidade do sistema com aplicações secundárias.....	8
7.2 Alocação dinâmica do cache ao criar <i>sequences</i> em bases SQL.....	8
7.3 Refatoração dos dados que são obtidos quando um novo <i>build</i> é gerado.....	8
8 Dificuldades e Limitações.....	10
9 Contribuições do estágio para o aluno.....	10
10 Relacionamento entre o Curso e o Projeto de Estágio-Emprego.....	10
11 Considerações sobre o Curso de Graduação.....	10
12 Trabalhos Futuros.....	11
13 Referências.....	11

1 Introdução

Este relatório tem como função apresentar os projetos desenvolvidos pelo autor durante o período do primeiro semestre de 2022 em seu ambiente de trabalho na empresa Philips. Os projetos apresentados no plano de trabalho no início do semestre aqui serão abordados de forma mais detalhada, documentando o processo de idealização dos projetos e como os mesmos foram desenvolvidos.

Em conjunto também serão abordados temas como DevOps, que se apresenta como uma área que entrega automações para processos da empresa, realizando melhorias que tem grandes resultados no valor da empresa, como por exemplo, a qualidade do seu produto e a agilidade em entregas (DAVIS; DANIELS, 2016).

Os projetos abordados focam principalmente no processo de entrega de artefatos para os clientes, com o primeiro projeto resultando em um auxílio ao implantador do sistema no cliente quando esse precise instalar ou atualizar alguma aplicação integrada ao sistema principal, o segundo projeto sendo a dinamização da criação de *sequences* em bases de dados SQL e o terceiro projeto sendo a validação de dados que são obtidos ao gerar um *build* de uma versão do sistema.

2 Empresa onde o estágio/emprego foi realizado

A Philips é uma empresa holandesa, fundada em 1891 por Frederik Philips e seu filho Gerard, inicialmente com foco na produção de lâmpadas elétricas. A empresa sucedeu em seu ramo a ponto de criar centros de pesquisa para poder apresentar inovações para o mercado e assim aumentar sua relevância histórica. Com o passar dos tempos a sua linha de produtos foi ficando mais abrangente, indo de lâmpadas para aparelhos de barbear, televisões e entre outros.

Um dos ramos que a Philips mais encontrou espaço para evoluir, foi o espaço de saúde, inicialmente com seus produtos de higiene pessoal, a Philips notou que esse era um mercado de possível investimento, logo focando em criar produtos tanto para saúde pessoal quanto para hospitais. Assim definindo uma das metas principais para a empresa, a de melhorar a vida de 2.5 bilhões de pessoas até 2030.

O seu portfólio de produtos no ramo da saúde é gigantesco e um desses é o Tasy, sistema inicialmente criado pela Wheb Sistemas, empresa brasileira que foi comprada pela Philips em 2010. O produto Tasy já era uma referência no mercado

de saúde nacional, e com esse compra a possibilidade de virar referência no mercado mundial poderia se tornar realidade.

A sede em Blumenau é focada exclusivamente no desenvolvimento e suporte ao Tasy, mas, mesmo assim, já é possível encontrar o Tasy em vários locais do mundo, possuindo até funcionários em outros países, como Índia e Estados Unidos.

3 Setor da empresa onde o estágio/emprego foi realizado

A Philips divide o setor de desenvolvimento em diversas áreas, parte focada no sistema hospitalar e sua regra de negócio e parte focada em entregas internas para ferramentas do dia a dia da empresa. Os projetos ocorreram dentro de uma das equipes que foca em entregas internas, essa sendo a equipe denominada como EMR DevOps, que é responsável principalmente pela geração de novas versões do sistema, manutenção dos repositórios de código fonte e ferramenta de atualização do sistema em cliente. Atualmente a equipe possui 17 integrantes e esses se dividem entre vários grupos, como os grupos focados em performance do sistema, plataforma e operação. Os projetos foram realizados na equipe de operação, o autor durante o período dos projetos atuava como desenvolvedor júnior em regime CLT, a equipe possui contato diário com os times de desenvolvimento, testes e suporte da Philips, respondendo dúvidas sobre processos e ferramentas e também auxiliando com possíveis inconformidades encontradas em rotinas utilizadas por esses times.

4 Período em que o estágio/emprego foi realizado

O estágio do aluno se deu durante o período de junho de 2020 até março de 2021, após essa data foi contratado na empresa como funcionário efetivo com o cargo de desenvolvedor júnior.

O projeto abordado teve como data de início 04 de abril de 2022 e tem data de encerramento estipulada para 03 de junho de 2022, assim o texto focará no que foi desenvolvido durante esses 2 meses de acompanhamento.

5 Métodos, Técnicas e Tecnologias Utilizadas

5.1 Métodos

O método utilizado para mensurar as entregas dos projetos foi o *SCRUM*, o *SCRUM* é um método que foi criado usando o *Agile* como base. *Agile* sendo um conjunto de valores e princípios para um desenvolvimento de softwares de forma iterativa e incremental (BECK *et al.*, 2001).

Logo o *SCRUM* possui um funcionamento focado em criar uma lista de recursos necessários a serem desenvolvidos em um sistema. Esses recursos são organizados pela importância dos mesmos, assim o time prioriza os recursos mais importantes primeiro (RUBIN, 2012).

Essa priorização é muito importante para alguns dos princípios do *SCRUM* que são seguidos ao aplicar o método, esses sendo (HEATH, 2021):

- Entregas são feitas em iterações curtas chamadas de *sprints*;
- Por ter recursos bem definidos, esses podem ser analisados a qualquer momento procurando por oportunidades de melhorias;
- O *SCRUM* por ser um método simples de se ler permite que as partes interessadas do projeto sigam o progresso do projeto.

Assim o método *SCRUM* é de grande valia para os projetos desenvolvidos na empresa, por permitir as entregas mais ágeis ao cliente e que sejam fáceis de alterar se necessário em algum momento.

5.2 Técnicas

Para a técnica de desenvolvimento, utiliza-se a documentação do que será desenvolvido junto com a criação de casos de testes iniciais que deverão ser validados sempre que uma alteração no projeto for realizada, conforme o desenvolvimento ocorre, mais testes podem ser criados para abordar pontos não pensados durante a concepção do documento inicial, mas o mesmo não se diz sobre a alteração dos testes já existentes, onde os mesmos só podem ser alterados se alguma parte da regra de negócio do projeto planejado foi alterado pelas partes interessadas.

5.3 Tecnologias

As tecnologias utilizadas durante os projetos são as seguintes:

- PL/SQL: O PL/SQL é uma linguagem de programação que permite facilitar a

forma como é trabalhado com tabelas e bases em SQL, permitindo a criação de funções, procedimentos, gatilhos e pacotes que auxiliam na reutilização de código(FEUERSTEIN; PRIBYL, 2015).

- Java: A linguagem de programação Java é amplamente difundida por ser uma das linguagens orientadas a objetos de maior facilidade de aprendizado, pois a mesma possui uma sintaxe fácil de entender e também por permitir que suas aplicações rodem em qualquer dispositivo, pois os códigos em Java são compilados para rodarem em máquinas virtuais próprias, permitindo que o programador não se preocupe em ter que criar códigos diferentes para vários dispositivos(SIERRA; BATES, 2005).
- Quarkus: Quarkus é um *framework* para ambientes em nuvem que realiza a integração de programas em Java com o Kubernetes, esse sendo uma plataforma de implantação de aplicações. O Quarkus permite que essa integração seja feita de forma mais simples e também apresenta ferramentas que facilitam o desenvolvimento para o programador, como redução do uso de memória e um tempo de inicialização da aplicação mais ágil(BUENO; PORTER, 2020).
- Jenkins: O Jenkins funciona como uma ferramenta para entrega contínua de aplicações, permitindo a automatização de *builds* e operações necessárias para entrega do produto(PARK; ADITHYA; GLESKE, 2021).

6 Projetos em que o aluno atuou durante o semestre

Durante o período avaliado, foram realizados 3 projetos diferentes, esses sendo:

- Criação de *Endpoint* para validar compatibilidade entre aplicações secundárias e sistema principal;
- Alocação dinâmica do cache ao criar *sequences* em bases SQL;
- Refatoração dos dados que são obtidos quando um novo *build* é gerado.

7 Descrição das Atividades Realizadas

Das atividades previstas no plano de trabalho, somente o projeto do Endpoint foi finalizado e entregue, pois esse possuía alta prioridade de entrega, o segundo

projeto referente a uma tela para clientes receberem pacotes de tradução não foi realizado pois outras atividades apareceram que possuíam um grau maior de importância de serem realizados.

7.1 Criação de *Endpoint* para compatibilidade do sistema com aplicações secundárias

O primeiro projeto trabalhado foi a criação de um *endpoint* que informaria a versão compatível de uma aplicação secundária com o sistema primário. Esse projeto foi necessário pois existem algumas aplicações secundárias que clientes podem instalar opcionalmente, mas elas possuem requisitos de compatibilidade com o sistema principal pois utilizam a mesma base de dados.

Assim foi montado esse *endpoint* utilizando o Java como linguagem de programação e o Quarkus como *framework* para implantação do mesmo, para as consultas de qual versão é compatível com o sistema, foi utilizado *querys* em SQL que consultavam tabelas presentes na base do cliente.

O *endpoint* recebe 3 parâmetros para a consulta, esse sendo a versão do sistema principal, o nome da aplicação e o build da versão. Com esses 3 parâmetros a consulta SQL verificava a tabela e retornava como resultado as versões da aplicação compatíveis, essas podendo ser versões exatamente compatíveis com os parâmetros enviados ou a maior versão dessa aplicação existente na tabela.

A figura 1 exibe o diagrama de atividades que o *endpoint* realiza, sendo essas atividades as validações necessárias que executam antes de retornar as informações que o usuário deseja.

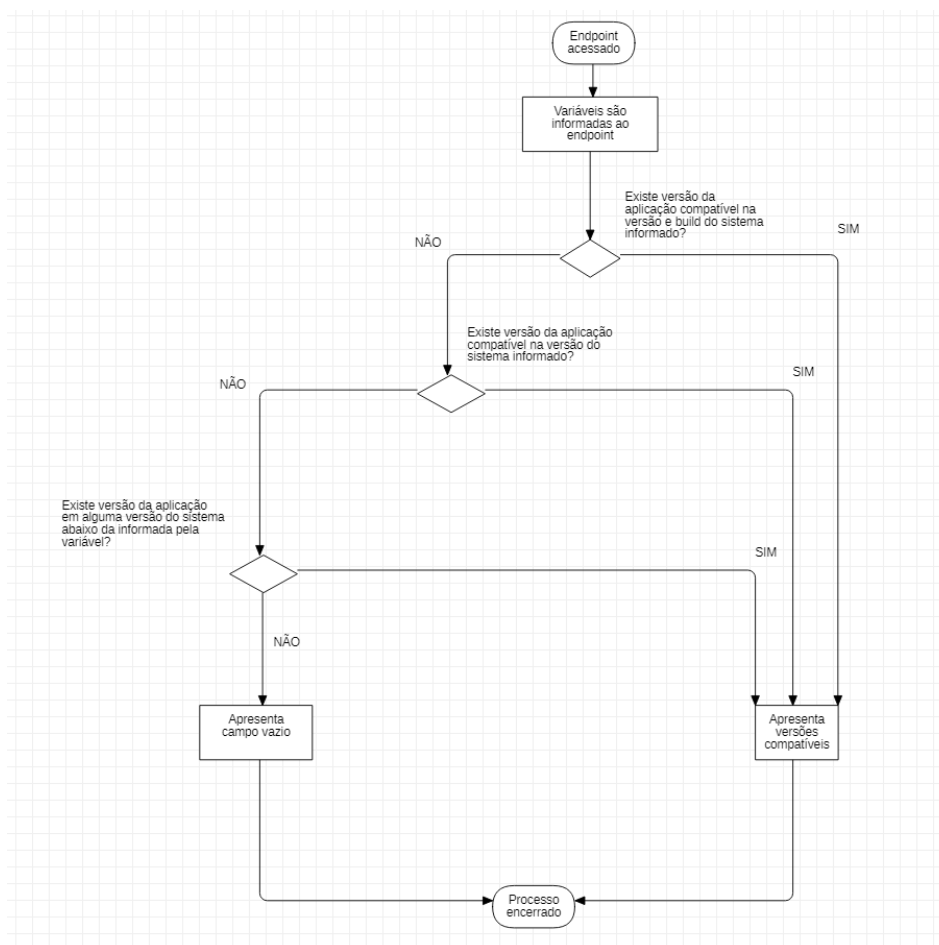


Figura 1: Diagrama de Atividades

O projeto organiza seus arquivos em 4 diferentes componentes, o objeto com as variáveis que serão utilizadas como parâmetros e retornos, o *resource* que apresenta as informações obtidas da consulta no *endpoint*, o *service* que realiza a integração da aplicação com a base e realiza a consulta e a classe de testes unitários que armazena os testes criados e é sempre executada quando alguma alteração é feita no código fonte.

O aprendizado e desenvolvimento do *endpoint* levou cerca de 2 semanas, por inicialmente ser necessário entender a utilização do Quarkus e como o mesmo integra o Java com o Kubernetes para criarem os contêineres da aplicação. Após isso foi necessário criar os componentes necessários para o funcionamento da aplicação. Com os componentes criados e alguns testes iniciais realizados, a aplicação passou por um tempo de maturação, sendo realizado testes pelo time que utilizará ele diariamente e assim encontrando possíveis alterações ou melhorias na aplicação.

Com o projeto já homologado pelas partes interessadas, foi realizado o lançamento dele, permitindo que a aplicação fosse utilizada de forma oficial pelo time que solicitou o mesmo.

7.2 Alocação dinâmica do cache ao criar *sequences* em bases SQL

O segundo projeto apareceu como uma oportunidade de melhoria na performance do uso de *sequences* em bases de dados SQL, que são criadas durante a atualização do sistema em suas bases de dados.

Como apresentado na figura 2, a criação de *sequence* se dava pela informação do campo cache em seu modo nulo('NoCache'), informar esse campo trazia alguns problemas de performance para clientes que possuíam *sequences* maiores em seu sistema, fazendo com que problemas de lentidão aparecessem.

```
CREATE SEQUENCE supplier_seq  
  MINVALUE 1  
  START WITH 1  
  INCREMENT BY 1  
  NOCACHE;
```

Figura 2: Exemplo de uma *sequence* criada com o NoCache

Para resolver esse problema de performance foi analisado uma forma de criar as *sequences* de forma dinâmica, para isso foi alocado um campo que informa o tamanho do cache para que quando o sistema fosse atualizado, o mesmo consultasse esse campo e verificasse se seria possível criar a *sequence* com cache pré-alocado.

Uma das situações necessárias de serem validadas era de quando o campo do tamanho do cache estivesse informado como nulo ou com um valor menor que 2, nesses casos a *sequence* deveria obrigatoriamente ser criada com o comando *NoCache*.

A atualização do sistema segue um *script* pré-montado em PL/SQL que realiza consultas para verificar quais tabelas e campos precisam ser criados ou atualizados, logo foi necessário alterar o *script* para que o mesmo obtivesse o valor informado no campo cache e com esse valor validar se a *sequence* deveria ser

criada com o comando *NoCache* informado ou se deveria utilizar o valor do campo, assim alterando o fim do comando de criação de sequência, abaixo na figura 3 é apresentado o exemplo de uma *sequence* criada com cache pré-alocado.

```
CREATE SEQUENCE supplier_seq  
  MINVALUE 1  
  START WITH 1  
  INCREMENT BY 1  
  CACHE 20;
```

Figura 3: Exemplo de uma *sequence* criada com cache informado

7.3 Refatoração dos dados que são obtidos quando um novo *build* é gerado

O projeto apareceu ao validar que os dados dos arquivos gerados pelo *build* de uma versão estavam sendo preenchidos incorretamente, especificamente a data da geração do *build*. Essa informação é apresentada ao cliente como uma informação do sistema e também é utilizado pelos times internos para fins de regulamentação de processos da empresa.

Inicialmente foi analisado quais eram os possíveis locais onde essa informação é gerada, o primeiro local era a própria tela que apresenta a informação ao cliente, foram feitas algumas investigações mas nada foi encontrado que já não estivesse sendo abordado. Após isso foi realizado uma investigação nos arquivos que são empacotados nos executáveis do sistema a fim de procurar alguma informação possível de ser utilizada.

Nesses arquivos foi encontrado alguns locais onde a data não estava sendo atualizada ao gerar o *build*, assim após a análise encontrar uma possível causa, deu-se início ao desenvolvimento de uma solução. Para a solução foi criado uma expressão regular que procurava pelos campos de data dentro dos arquivos, assim podendo mudar eles para a data correta.

Para os testes, foi criado uma *pipeline* de teste da geração dos *builds* utilizando o Jenkins como ferramenta de automatização, onde o comando de alteração das datas foi informado dentro de um dos processos executados pela *pipeline*. Com essa alteração informada, os testes se iniciaram com *builds* não oficiais sendo gerados e os arquivos sendo validados para garantir que a data foi alterada corretamente, durante o processo de teste foram realizadas algumas

alterações na expressão regular pois alguns casos de teste não estavam apresentando resultados positivos. Assim, ao encontrar o resultado final correto, o mesmo foi colocado em teste em alguns arquivos gerados de forma oficial, essas validações ainda estão ocorrendo e quando as mesmas tiverem sucesso serão enfim integradas em toda a *pipeline* oficial.

8 Dificuldades e Limitações

Durante o desenvolvimento dos projetos algumas dificuldades foram encontradas, tanto no desenvolvimento do mesmo quanto em situações externas que necessitavam de uma atuação imediata, obrigando assim a parar o desenvolvimento dos mesmos.

Em relação ao primeiro projeto, uma das maiores dificuldades foi todo o processo de aprendizado da utilização do Quarkus para o desenvolvimento do *endpoint*, alguns dias de estudos foram utilizados para entender a documentação inicial da ferramenta e após isso começar a desenvolver o projeto.

Para o segundo projeto a maior dificuldade foi encontrada em limitações da própria Oracle, que não permite a criação de *sequences* com o valor de cache igual a 1 ou 0, assim foi necessário pensar em possíveis validações que permitiam a alteração do comando para informar o cache somente se o valor que era utilizado fosse maior ou igual a 2.

O terceiro projeto teve como dificuldade principal o de encontrar da onde que o sistema obtia a informação da data de geração do *build*, isso continuou como uma dificuldade até o ponto onde foi acordado que o melhor seria procurar nas rotinas que geram esse *build* onde possivelmente deveria ocorrer essa atualização de informação.

9 Contribuições do estágio para o aluno

As contribuições que o estágio realizado na Philips ofereceram foram de compreender uma nova segmentação dentro da T.I. que é o *DevOps*, assuntos que abordam essa área são de grande interesse por se relacionarem principalmente com entregas contínuas e de qualidade das aplicações, que atualmente são grandes diferenciais para um sistema.

10 Relacionamentos entre o Curso e o Projeto de Estágio-Emprego

O curso auxiliou durante o projeto por apresentar a base necessária para a programação dos projetos, tanto do desenvolvimento do código em Java quanto na modelagem de tabelas em SQL.

Junto com a base de programação, também apresentou uma base que foi utilizada quando o método de *SCRUM* foi apresentado para as entregas que seriam realizadas, o conhecimento prévio obtido durante o curso auxiliou quando o método foi integrado ao time.

11 Considerações sobre o Curso de Graduação

O curso apresentou uma ótima base tanto na área de desenvolvimento quanto na área de análise de sistemas, permitindo que a curva de aprendizado durante as experiências profissionais fosse suavizado, possibilitando um aprendizado rápido das atividades realizadas dentro da empresa.

Como oportunidade de melhoria seria válido apoiar o uso de boas práticas de programação para os alunos do início do curso até o fim, assim possibilitando que tenham esse diferencial entre os outros novos profissionais da área. Além desse diferencial é importante salientar que saber a diferença entre um código bem escrito do que não é desenvolve uma interpretação de código crítica que alavanca a carreira profissional do aluno, pois o mesmo pode encontrar soluções para problemas que sejam relacionados a códigos mal desenvolvidos.

12 Trabalhos Futuros

Com o desenvolvimento do projeto e a evolução própria que ocorreu durante o curso, é esperado que continue a estudar sobre novos assuntos na área a fim de se tornar um melhor profissional, buscando sempre a excelência no trabalho realizado.

13 Referências

BECK, Kent; BEEDLE, Mike; VAN BENNEKUM, Arie; COCKBURN, Alistair; CUNNINGHAM, Ward; FOWLER, Martin; GRENNING, James; HIGHSMITH, Jim; HUNT, Andrew; JEFFRIES, Ron; KERN, Jon; MARICK, Brian; MARTIN, Robert C; MELLOR, Steve; SCHWABER, Ken; SUTHERLAND, Jeff; THOMAS, Dave. Agile Manifesto. 2001. **Software Development**. .

BUENO, Alex Soto; PORTER, Jason. **Quarkus Cookbook**. 1st ed. Sebastopol: O'Reilly Media, Inc., 2020. vol. 1, .

DAVIS, Jennifer; DANIELS, Katherine. **Effective DevOps**. [S. l.: s. n.], 2016. Available at: <https://www.oreilly.com/library/view/effective-devops/9781491926291/>.

FEUERSTEIN, Steven; PRIBYL, Bill. Oracle PL SQL Programming. **Oracle PL SQL Programming**, 2015. .

HEATH, Fred. **The Professional Scrum Master (PSM I) Guide**. 1st ed. Birmingham: Packt, 2021. vol. 1, .

PARK, Calvin Sangbin; ADITHYA, Lalit; GLESKE, Samuel. **Jenkins Administrator's Guide**. 1st ed. Birmingham: Packt Publishing Ltd., 2021. vol. 1, .

RUBIN, Kenneth S. Essential Scrum. **Essential Scrum: A Practical Guide to the Most Popular Agile Process**. [S. l.: s. n.], 2012.

SIERRA, Kathy; BATES, Bert. **Head First Java**. [S. l.: s. n.], 2005.