

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
() PRÉ-PROJETO	(x) PROJETO	ANO/SEMESTRE: 2023/1

ASSISTENTE DE ATUALIZAÇÃO DE VERSÃO DE PROJETOS ANGULAR

Nathan Reikdal Cervieri

Prof. Marcel Hugo - Orientador

1 INTRODUÇÃO

No cenário da internet é impossível determinar o impacto que páginas da web tem na nossa percepção da tecnologia de comunicação mundial. Tudo pode ser encontrado em páginas da web, desde conhecimentos diversos como a Wikipédia, ferramentas de pesquisa como o Google até websites de compra online como o Mercado livre e a Amazon. Websites são uma interface de comunicação com as várias facetas do mundo online e com isso temos a variedade de sistemas e projetos de código relevantes a seus domínios.

Para facilitar o desenvolvimento de páginas da web há vários frameworks de desenvolvimento. Um deles é o Angular, um framework orientado a componente que provê várias funcionalidades que auxiliam a criação de websites interativos, dinâmicos e escaláveis com esforço mínimo (ANGULAR, 2022). De acordo com uma pesquisa feita pela plataforma StackOverflow (2022), Angular está entre as cinco ferramentas mais utilizadas em desenvolvimento web, com aproximadamente 20% das quase 60 mil repostas ao formulário já terem usado a tecnologia.

O Angular, assim como outras ferramentas, funciona em um sistema de versão no qual novas versões trazem novas funcionalidades e correção de inconsistências ou problemas de segurança. A mais recente versão do Angular é a 15. Essas atualizações também trazem alterações que as tornam parcialmente incompatíveis com código escrito previamente. Assim, se o desenvolvedor que utiliza o framework não realizar o esforço para atualizar versões do sistema com frequência pode acabar em um estado em que o custo de atualizar todo o projeto para uma nova versão não parece valer a vantagem que as atualizações trazem. Isso é inaceitável em casos em que a falta de atualização causa falhas críticas de segurança, mas ainda pode ser complicado justificar o custo da atualização contra reescrever partes do sistema do início.

A partir destas afirmações, este projeto pretende desenvolver uma ferramenta que auxilia na atualização de aplicações Angular de modo a diminuir o custo e tempo de alteração. O propósito é diminuir a propensão a erros que refatoração de larga escala pode trazer a um projeto. Por meio da análise e alteração automática dos pré-requisitos, que podem ser encontrados no site que cataloga alterações necessárias entre versões do Angular (ANGULAR, 2023), se buscará garantir a funcionalidade entre a versão base e a versão alvo. O projeto além disso também fará um levantamento bibliográfico para concentrar conhecimento sobre o tema de atualização automatizada entre versões.

1.1 OBJETIVOS

O objetivo deste trabalho é simplificar o processo de atualização de código escrito na versão 14 para a 15 do framework Angular para diminuir o tempo e trabalho necessários através da automatização da análise do projeto e alterações a serem feitas.

Os objetivos específicos são:

- apontar mudanças necessárias para atualização entre versões 14 e 15 do Angular;
- realizar mudanças encontradas de maneira automática quando possível;
- orientar o desenvolvedor sobre mudanças apontadas que não foram possíveis de serem realizadas.

2 TRABALHOS CORRELATOS

Nesta seção são apresentados trabalhos com características semelhantes aos principais objetivos do estudo proposto. O primeiro é uma ferramenta de auxílio de atualização de projetos .net (DOTNET, 2021). O segundo é um assistente de atualização parecido com o primeiro, mas com foco em atualização de projetos AngularJS para Angular (OLSON, 2018). O terceiro é uma ferramenta que converte Python 2 para Python 3 (PYTHON, 2012).

2.1 DOTNET UPGRADE ASSISTANT

O Dotnet upgrade Assistant (DOTNET, 2021) é uma ferramenta *open source* desenvolvida inicialmente pela Microsoft para facilitar o processo de atualização de projetos e soluções .net de versões passadas para versões mais atualizadas. Atualmente, suporta os seguintes tipos de projeto: ASP.NET MVC, Windows Forms, Windows

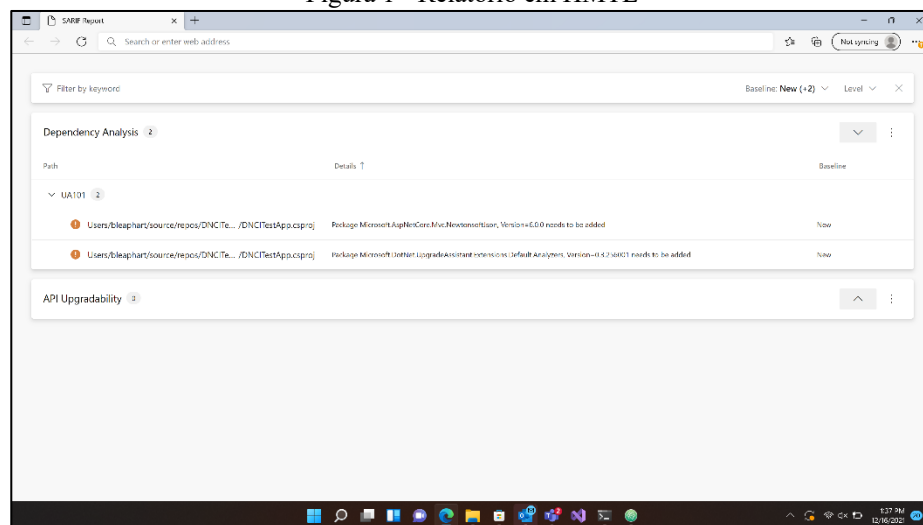
Presentation Foundation (WPF); Console app; Libraries; UWP to Windows App SDK (WinUI); Xamarin.Forms to .NET MAUI.

A ferramenta tem dois pontos principais: analisar e atualizar. Ao analisar um projeto, a ferramenta verifica pacotes, referências, referências a framework e chamadas a pacotes que possam ter alterações que causam quebra e APIs não suportadas, com o intuito de decidir uma ação entre remover, adicionar e atualizar as partes para ficar compatível com a versão de destino.

Ao atualizar o projeto, o Upgrade Assistant determina os projetos e a ordem que eles devem ser atualizados, atualiza o tipo de arquivo, remove pacotes NuGet desnecessários, muda o alvo do pacote para a versão desejada, atualiza os pacotes de acordo com o passo de análise, faz atualizações simples no código C# para garantir compatibilidade, adiciona códigos e partes simples para a versão de destino e adiciona analisadores para ajudar com a atualização, os analisadores criados no passo de update apontam alterações manuais que ainda são necessárias para garantir o funcionamento da solução.

Para utilizar o assistente, se usa uma interface por linha de comando em duas partes, uma parte interativa e outra não interativa. Durante a parte interativa, o usuário acompanha a ferramenta durante alguns trechos do processo em que tem que decidir como algumas ações serão realizadas. Em outros trechos, a ferramenta roda de maneira não interativa, realizando alterações de modo automático. Após todo processo, é gerado um arquivo, ou conjunto de logs que indica o processamento realizado, assim como gera um relatório dos passos como pode ser visto na Figura 1.

Figura 1 - Relatório em HTML



Fonte: Dotnet (2021).

A ferramenta não é capaz de fazer todos os passos por si só, requerendo interação do usuário em alguns casos, assim como necessita de autenticação para buscar pacotes NuGet relevantes. Atualmente o assistente de atualização não é capaz de analisar a viabilidade e/ou custo da atualização, ao afirmar que se assume que projetos utilizando o aplicativo já foram revisados.

2.2 NGMIGRATION ASSISTANT

O ngMigration Assistant (OLSON, 2018) é uma ferramenta de linha de comando que varre aplicações em AngularJS e recomenda sugestões para realizar a migração para Angular. A ferramenta analisa o código da aplicação e cria relatórios de antipadrões de desenvolvimento e quais arquivos e linhas de código necessitam ser alteradas. Também gera um relatório que indica recomendações e preparações necessárias para realizar a atualização. Para isso, tem-se alguns passos para realizar a análise: primeiro busca o caminho de todos os arquivos para serem analisados e quais arquivos para ignorar através de parâmetros de inicialização; depois conta assincronamente as linhas de código encontradas no projeto e executa o passo de análise que realiza vários testes para verificar as alterações necessárias. Finalmente, gera relatórios de anti-padrão e de recomendação para atualização do projeto.

O ngMigration Assistant realiza o trabalho de catalogar quais os pontos do projeto são necessários realizar algum tipo de alteração para se conseguir atualizar o projeto do AngularJS para Angular, porém não realiza nenhuma alteração de maneira automatizada. O relatório traz também uma contagem das linhas de código que necessitam alteração de maneira a estimar tempo de desenvolvimento, ao apontar que 880 linhas de código é equivalente a 1 mês de trabalho. Conclui-se que a ferramenta é útil para diminuir o trabalho de planejamento de migração, mas não auxilia o processo de atualização em si.

2.3 PYTHON 2 TO 3 CONVERTER

Python 2to3 converter é um programa de linha de comando escrito em Python criado para transformar código feito para Python 2 para Python 3. O código roda uma coleção de *fixers* para transformar em código Python 3 válido, e é bastante flexível a ponto de permitir ao usuário criar seus próprios para cobrir situações em que a ferramenta falhe (PYTHON, 2012). Apesar de ter sido desenvolvido em 2012, o código recebeu atualizações até ano passado, no momento está marcado para deixar de ser usado a partir da versão 3.13 do Python.

A ferramenta pode ser utilizada para converter arquivos individuais ou projetos inteiros e transcreve os arquivos para um novo arquivo ou diretório, porém com várias opções para modificar o comportamento padrão. Para casos onde a ferramenta não é capaz de alterar código, é feito um arquivo de log com os locais que ainda é necessário ação.

3 PROPOSTA DA FERRAMENTA

Nesta seção é realizada uma análise dos trabalhos correlatos, assim como uma descrição dos requisitos do trabalho e a exposição da metodologia prevista.

3.1 JUSTIFICATIVA

O Quadro 1 apresenta uma comparação entre os trabalhos correlatos descritos na seção 2, expondo suas características e relacionando um aos outros.

Quadro 1 - Comparativo dos trabalhos correlatos

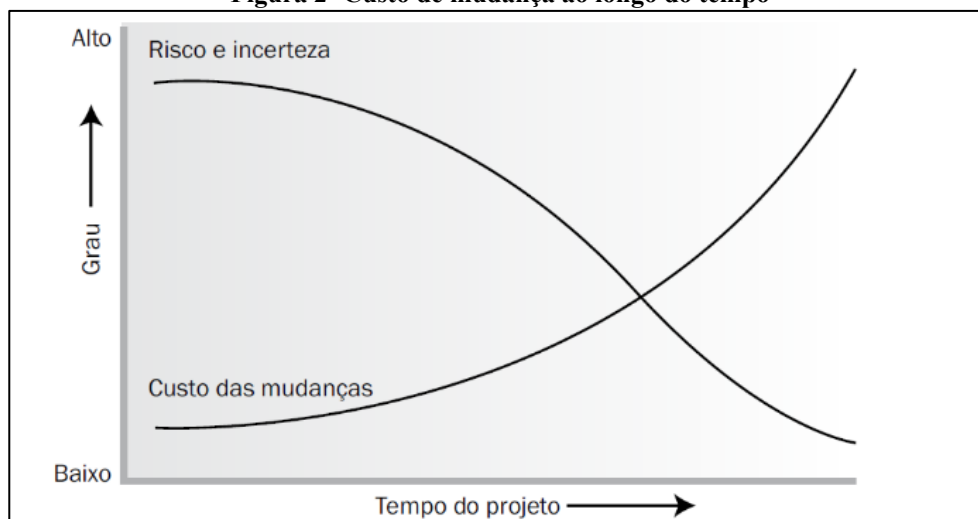
Trabalhos Correlatos Características	DONET UPGRADE ASSISTANT	NGMIGRATION ASSISTANT	PYTHON 2 TO 3 CONVERTER
Migra código	Entre versões e frameworks	Entre frameworks Angular	Entre código Python 2 para Python 3
Analisa código	Sim	Sim	Não
Sugere alterações necessárias	Sim	Sim	Não
Indica inconsistências entre versões	Sim	Sim	Não
Realiza alteração de código	Sim	Não	Sim

Fonte: elaborado pelo autor.

Todos os trabalhos correlatos visam a migração de código entre diferentes contextos. O Dotnet (2021) procura fazer essa migração entre frameworks ou versões do mesmo framework, enquanto Olson (2018) cataloga as alterações necessárias entre dois frameworks do angular e Python (2012) converte projetos Python 2 para Python 3. Os três trabalham com atualização de código para consistência entre versões, porém para contextos diferentes. Todos os correlatos fazem sugestões de quais alterações tem que ser feitas para concluir o processo, mas o conversor de Python (Python, 2012) não cria um relatório do que foi realizado ou tem que ser feito. A ferramenta ngMigration Assistant (OLSON, 2018) é a única que não faz alteração do código diretamente.

A existência de ferramentas e métodos para agilizar a atualização de código é importante para garantir a longevidade de projetos. Com o passar do tempo e manutenções sendo realizadas, sistemas têm sua qualidade degradada devido a vários desenvolvedores e padrões diferentes sendo aplicados. A partir dessa diminuição de qualidade se tem um aumento de custo de desenvolvimento, porém, com menos riscos, considerando que a estabilidade do sistema aumenta com o tempo do projeto, como pode ser visto na Figura 2.

Figura 2- Custo de mudança ao longo do tempo



Fonte: Pmbok (2017)

A ferramenta proposta visa auxiliar em mudanças de sistemas desenvolvidos na versão 14 do framework Angular com o intuito de diminuir custo e facilitar a manutenção para alcançar a versão 15. Como contribuição científica o projeto irá catalogar o processo de criação de uma ferramenta deste tipo assim como realizar uma análise da literatura sobre o assunto, para tornar o conhecimento mais acessível para desenvolvedores com necessidade de manter sistemas difíceis de atualizar. Como contribuição social, disponibiliza uma ferramenta gratuita que possa ser utilizada por desenvolvedores Angular para auxiliar seu processo de atualização entre versões. Ainda há a contribuição tecnológica, uma vez que todo o código do projeto desenvolvido estará disponível de forma aberta à comunidade de desenvolvedores.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A ferramenta deverá:

- a) permitir análise de viabilidade para conversão de projeto completo (Requisito Funcional - RF);
- b) permitir análise de viabilidade para conversão de arquivos singulares (RF);
- c) criar relatório de alterações manuais e automáticas (RF);
- d) alterar arquivos de maneira automatizada (RF);
- e) ser desenvolvida em Angular a interface de usuário (Requisito Não Funcional - RNF);
- f) ser desenvolvido em C# o servidor de análise e substituição (RNF);
- g) utilizar banco de dados não relacional para guardar alterações entre versões (RNF);
- h) utilizar regex para buscar pontos de alteração (RNF);
- i) disponibilizar projeto online (RNF).

3.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: buscar fontes bibliográficas relacionadas ao objetivo do estudo proposto, como análise de código, alteração automática de código e cálculo de esforço de desenvolvimento;
- b) elicitação dos requisitos: reavaliar os requisitos funcionais e não funcionais, com o objetivo de atender o estudo proposto;
- c) especificação do trabalho: elaborar diagrama de classes e diagrama de casos de uso;
- d) definir método de catalogação: definir objeto de controle de etapa e maneira de guardar esses dados;
- e) catalogar alterações necessárias: utilizar o site que lista alterações em *updates* do Angular para catalogar as alterações necessárias entre versões;
- f) desenvolver primeira versão da interface visual: desenvolver uma interface visual em HTML usando framework Angular para importar arquivo e receber resposta;
- g) desenvolver servidor para análise arquivo e substituição: criar um módulo servidor implementado a lógica que utilize as ações catalogadas e busque código incompatível nos arquivos utilizando regex e os substitua;
- h) criar relatórios de alterações manuais necessárias e automáticas realizadas: retornar lista de alterações que não podem ser realizadas automaticamente e quais alterações foram feitas;
- i) desenvolver versão final da interface visual: alterar interface para permitir importar mais de um arquivo por vez, assim como demonstrar os relatórios produzidos em tela de maneira compreensiva para o usuário;

- j) teste: buscar ou criar projetos em Angular 14 para testar a funcionalidade esperada na proposta;
- k) validação: utilizar os projetos de teste e realizar alterações de maneira manual com intuito de comparar o esforço necessário;
- l) publicar projeto para acesso público: publicar o código do projeto e interface para acesso público.

As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma

etapas / quinzenas	2023									
	jul.		ago.		set.		out.		nov.	
	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
elicitación dos requisitos										
especificação do trabalho										
definir método de catalogação										
catalogar alterações necessárias										
desenvolver primeira versão da interface visual										
desenvolver método de análise arquivo e substituição										
criar relatórios de alterações manuais necessárias e automáticas realizadas										
desenvolver versão final da interface visual										
teste										
validação										
publicar projeto para acesso público										

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Neste capítulo serão discutidos os conceitos que fundamentarão o trabalho como um todo, começando pelo Angular (ANGULAR, 2022) e seu guia de atualização de versão (ANGULAR, 2023); uma análise de ferramentas de atualização de código; e finalmente uma visão de testes de software.

4.1 ANGULAR

Frameworks de desenvolvimento são uma coleção de ferramentas e bibliotecas de código cuja função é alcançar um objetivo técnico ou tornar o desenvolvimento de aplicações mais fácil (MAURICIO, 2018). O Angular é um framework baseado em typescript (ANGULAR, 2022) criado em 2010 que auxilia na construção de projetos web através de um conjunto robusto de ferramentas e de funcionalidades importantes para o desenvolvimento de aplicações como: roteamento, injeção de dependência e vinculação de dados bidirecional (MILLER, 2021). Permite a construção de aplicações de pequeno e grande porte e facilita a criação de projetos escaláveis de forma rápida e consistente ao remover a responsabilidade de criação de componentes básicos e estrutura de comunicação entre esses componentes dos desenvolvedores (ANGULAR, 2022); (MILLER, 2021).

De acordo com uma entrevista com o criador do Angular (COLUMBUS, 2016), o projeto começou como um trabalho alternativo na Google para facilitar o desenvolvimento interno em plataformas web, eventualmente sendo liberado para todos, não apenas desenvolvedores Google. Hoje o Angular é utilizado por milhares de usuários, como indicado por uma pesquisa feita pelo website Stackoverflow (2022), que aponta como a quinta tecnologia ou *framework* mais utilizado para desenvolvimento web, totalizando aproximadamente onze mil seleções no questionário. Entre as companhias que utilizam o Angular se pode encontrar: a Google, Microsoft, Netflix, Paypal, etc. (QUINN, 2022).

Entre as informações disponíveis para o framework Angular se encontra o Angular Update Guide (ANGULAR, 2023). Essa página da web permite ao usuário indicar quaisquer duas versões do Angular para receber um guia listando as alterações gerais necessárias para adequar da versão mais antiga para mais nova. O guia também permite escolher a complexidade da aplicação, medida em: básica, média e avançada (ANGULAR, 2023). Também é possível escolher entre algumas dependências que interferem na adequação entre versões, sendo elas: o uso de ngUpgrade para combinar AngularJS e Angular, o uso de Angular Material e se o usuário utiliza sistema operacional Windows (ANGULAR, 2023) para assim permitir atualizar projetos que estão em situações diversas.

4.2 FERRAMENTA DE ATUALIZAÇÃO DE CÓDIGO

Uma prática da programação é a utilização de versionamento, que consiste em ter diversas versões de um programa trabalhando de maneira cronológica. Para nominar ou referenciar cada uma destas diferentes versões criou-se um método denominado *Semantic Versioning* (semver), cuja versão mais recente é a 2.0.0, liberada em 2013 (SEMVER, [2013]). O método utiliza a noção de versão *MAJOR.MINOR.PATCH*. Por trás destes nomes se tem que a alteração de versão *major* é quando há uma alteração que quebra funcionalidades na aplicação ou

biblioteca, *minor* é quando uma função é adicionada de maneira retrocompatível e *patch* para quando se tem uma alteração que é correção de problemas, ainda retrocompatível (SEMVER, [2013]).

Versões *major*, que causam quebras, são o que torna necessário a alteração do código entre versões de linguagens ou frameworks de código. Em alguns casos é necessário realizar essas mudanças de maneira manual, seguindo um guia de instruções ou uma lista de mudanças de quebra, como pode ser encontrando no Angular Update Guide (ANGULAR, 2023) ou o ngMigration Assistant (OLSON, 2019). Outros casos existe uma ferramenta para auxiliar nesse processo, realizando alterações automatizadas como o .NET Upgrade Assistant (DOTNET, 2021) ou o Python2to3 (PYTHON, 2012).

Para as ferramentas que atuam diretamente em código, é feito um processo de busca no código fonte com o intuito de buscar locais que devem ser alterados e como fazer essas alterações. Todas ferramentas utilizam um tipo de expressão regular (regex) para encontrar as partes do código inválidas e depois aplicam um código de substituição simples, ou criam um tipo de relatório para informar que uma alteração é necessária.

Expressões regulares são padrões usados para buscar tipos de texto em strings (MDN, 2023) em várias linguagens de programação. Friedl (2006) diz que expressões regulares são a chave para processamento de texto flexível, potente e eficiente. A ferramenta proposta por este projeto trabalha com processamento de texto e substituição de texto na forma de código, sendo essencial a utilização de regex.

4.3 TESTES

De acordo com Dias Neto (2007), “teste de software é o processo de execução de um produto para determinar se ele atingiu suas especificações...”. Testes no desenvolvimento em cascata são feitos de maneira manual, mas em com desenvolvimento ágil, são feitos de maneira automatizada (VALENTE, 2022).

Testes de software são um passo integro ao desenvolvimento de ferramentas modernas. Métodos como o Test-Driven Development (TDD) aplicam uma prática onde o teste automatizado é escrito “... pelo desenvolvedor, mesmo antes que as funcionalidades sejam implementadas” (CAMPOPIANO, 2022). Crispin e Gregory (2009) afirmam que esses testes unitários permitem ao programador escrever código com segurança que não causará impactos inesperados ao sistema. Em uma apresentação feita em Kirkland, Chirila (2014) apresentou que na empresa Google eles buscam ter uma “cobertura de teste”, quantidade de linhas testadas por teste unitário, de aproximadamente 85% de todo código, mas isso nem sempre é possível, com a quantidade sendo variável com base na linguagem de programação do sistema.

Existem 5 tipos de teste de software, de acordo com Rocha (2001, apud DIAS NETO, 2007): unidade; integração; sistema; aceitação e regressão. Teste de unidade ou teste unitário “tem por objetivo explorar a menor unidade de projeto, [...] os métodos dos objetos ou mesmo pequenos trechos de código” (DIAS NETO, 2007). Teste de integração “visa provocar falhas associadas às interfaces entre os módulos...” (DIAS NETO, 2007). Teste de sistema executa e “avalia o software em busca de falhas por meio da utilização do mesmo...” (DIAS NETO, 2007). Teste de aceitação “são realizados geralmente por um restrito grupo de usuários finais do sistema.” (DIAS NETO, 2007) dessa maneira agindo como um teste de sistema generalizado. Teste de regressão é relevante após a entrega, durante o tempo de atualização ao “aplicar a cada nova versão do software [...] todos os testes que já foram aplicados nas versões [...] anteriores do sistema” (DIAS NETO, 2007).

O teste unitário será utilizado durante todo processo de desenvolvimento de código do trabalho proposto. Teste de integração após o desenvolvimento de uma interface visual para garantir o funcionamento da mesma com o módulo servidor. Teste de sistema para validar o resultado do projeto e determinar conclusões. Para o intuito e escopo da ferramenta proposta deste projeto, não será aplicado teste de aceitação ou teste de regressão.

REFERÊNCIAS

- ALTEXSOFT. **The Good and the Bad of C# Programming**. 2021. Disponível em: <https://www.altexsoft.com/blog/c-sharp-pros-and-cons/>. Acesso em: 20 abr. 2023.
- ANGULAR. **What is Angular**. 2022. Disponível em: <https://angular.io/guide/what-is-angular>. Acesso em: 16 abr. 2023.
- ANGULAR. **Angular Upgrade Guide**. 2023. Disponível em: <https://update.angular.io/>. Acesso em: 20 abr. 2023.
- CAMPOPIANO, Giovanni Teixeira. **Importância de testes unitários no desenvolvimento web**. 2022. Trabalho de conclusão de curso (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) - Faculdade de Tecnologia de São Paulo, São Paulo, 2022.
- CHIRILA, Andrei. **Test coverage at Google**. 2014. Disponível em: <https://developers.google.com/google-test-automation-conference/2014/presentations>. Acesso em: 17 jun. 2023.

COLUMBUS, Louis. **Misko Hevery, Inventor of Angular And How Open Source Languages Are Redefining Enterprise Software**. 2016. Disponível em: <https://www.forbes.com/sites/louiscolumbus/2016/11/14/misko-hevery-inventor-of-angular-and-how-open-source-languages-are-redefining-enterprise-software/?sh=3d699b85270d>. Acesso em: 18 jun. 2023.

COSTA, Breno Gustavo Soares da. **Uma proposta de migração de sistemas legados do governo para a nuvem**. 2018. 110 f. Dissertação (Mestrado) - Curso de Computação Aplicada, Departamento de Ciência da Computação, Universidade de Brasília, Brasília, 2018. Disponível em: <https://repositorio.unb.br/handle/10482/34321>. Acesso em: 20 abr. 2023.

CRISPIN, Lisa; GREGORY, Janet. **Agile Testing: a practical guide for testers and agile teams**. Boston: Pearson Education, 2009. 533 p.

DIAS NETO, Arilo Cláudio. Introdução a Teste de Software. **Engenharia de Software Magazine**, São Paulo, v. 1, n. 7, p. 54-59, 2007.

DOTNET. **.NET Upgrade Assistant**. 2021. Disponível em: <https://github.com/dotnet/upgrade-assistant>. Acesso em: 20 abr. 2023.

FRIEDL, Jeffrey E. F.. **Mastering Regular Expressions**. 3. ed. Sebastopol: O'reilly Media, Inc., 2006.

MAURICIO, Jordan. **Understanding Web Development Frameworks in 2018: the most important frontend libraries to know as a web developer**. The most important frontend libraries to know as a web developer. 2018. Disponível em: <https://levelup.gitconnected.com/web-development-frameworks-475e06016093>. Acesso em: 18 jul. 2023.

MDN. **Regular expressions**. 2023. Disponível em: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_expressions. Acesso em: 18 jun. 2023.

MILLER, Stephan. **What Is Angular?** 2021. Disponível em: <https://www.codecademy.com/resources/blog/what-is-angular/>. Acesso em: 20 abr. 2023.

OLSON, Ella. **NgMigration Assistant**. 2018. Disponível em: <https://github.com/ellamaolson/ngMigration-Assistant>. Acesso em: 17 jun. 2023.

PMBOK. Project Management Institute (ed.). **Um Guia do Conhecimento em Gerenciamento de Projetos: guia pmbok**. 6. ed. Newtown Square, Pensilvânia, EUA: Project Management Institute, 2017. 756 p.

PYTHON. **2to3: automated python 2 to 3 code translation**. Automated Python 2 to 3 code translation. 2012. Disponível em: <https://docs.python.org/3/library/2to3.html>. Acesso em: 12 jun. 2023.

QUINN, Nicola. **Biggest Companies Keeping Angular Popular**. 2022. Disponível em: <https://pangea.ai/blog/tech-stack/biggest-companies-keeping-angular-popular>. Acesso em: 17 jun. 2023.

SEMVER. **Semantic Versioning 2.0.0**. [2013]. Disponível em: <https://semver.org/>. Acesso em: 18 jun. 2023.

STACKOVERFLOW. **2022 Developer Survey**. 2022. Disponível em: <https://survey.stackoverflow.co/2022/>. Acesso em: 20 abr. 2023.

VALENTE, Marco Tulio. **Engenharia de Software Moderna: princípios e práticas para desenvolvimento de software com produtividade**. Belo Horizonte: Independente, 2020. 408 p.