Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Engenharia de Software

# Desenvolvimento de Software Seguro: propostas para a melhoria da formação dos Engenheiros de Software nas universidades brasileiras

Autor: Fernando Miranda Calil
Orientadora: Profa. Dra. Elaine Venson

Brasília, DF
2023

Fernando Miranda Calil

# Desenvolvimento de Software Seguro: propostas para a melhoria da formação dos Engenheiros de Software nas universidades brasileiras

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientadora: Profa. Dra. Elaine Venson

Brasília, DF

2023

Fernando Miranda Calil

Desenvolvimento de Software Seguro: propostas para a melhoria da formação dos Engenheiros de Software nas universidades brasileiras/ Fernando Miranda Calil. – Brasília, DF, 2023-

114 p. : il. (algumas color.) ; 30 cm.

Orientadora: Profa. Dra. Elaine Venson

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2023.

1. Segurança de software. 2. Desenvolvimento seguro de software. I. Profa. Dra. Elaine Venson. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Desenvolvimento de Software Seguro: propostas para a melhoria da formação dos Engenheiros de Software nas universidades brasileiras

CDU 02:141:005.6

Fernando Miranda Calil

# Desenvolvimento de Software Seguro: propostas para a melhoria da formação dos Engenheiros de Software nas universidades brasileiras

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 19 de dezembro de 2023:

---

**Profa. Dra. Elaine Venson**
Orientadora

---

**Prof. Dr. John Lenon Cardoso Gardenghi**
Convidado 1

---

**Prof. Dr. Glauco Vitor Pedrosa**
Convidado 2

Brasília, DF
2023

# Resumo

Em um mundo tão dependente de produtos de software, a segurança precisa ser uma característica fundamental da aplicação, principalmente em situações que utilizam dados sensíveis de seus usuários. Para que o desenvolvimento de aplicações possa atender aos requisitos de segurança exigidos atualmente, é importante desenvolver o produto planejando e implementando a segurança em cada passo de construção. No entanto, o desenvolvimento de software seguro requer conscientização, conhecimento e capacitação dos profissionais e este conteúdo ainda não é oferecido em boa parte das instituições formadoras de Engenheiros de Software brasileiras. Este trabalho propõe-se a investigar o cenário educacional da Segurança de Software por meio da análise e comparação dos currículos de universidades brasileiras com universidades internacionais e modelos de estrutura curricular propostos por pesquisas da área, bem como avaliar a percepção de discentes a respeito deste tema. Para isso, será realizada uma pesquisa documental, análise de lacunas dos currículos e um *survey* com estudantes da Engenharia de Software da *Universidade de Brasília* (UnB). A partir das informações coletadas, serão propostas recomendações para tratar o tema do Desenvolvimento Seguro de Software na formação dos estudantes no curso de Engenharia de Software da UnB.

**Palavras-chaves**: segurança em aplicações de software; desenvolvimento seguro; formação de profissionais; Engenharia de Software; universidades brasileiras; estrutura curricular; lacunas; treinamento.

# Abstract

In a world so dependent on software products, security needs to be a fundamental feature of the application, especially in situations that use sensitive data from its users. So that the development of applications can meet the security requirements demanded today, it is important to develop the product to plan and implement security in each construction step. However, the development of secure software requires awareness, knowledge and training of professionals and this content is still not offered in most institutions that train Brazilian Software Engineers. This work proposes to investigate the educational scenario of Software Security through the analysis and comparison of the curricula of Brazilian universities with international universities and models of curricular structure proposed by researchers in the area, as well as to evaluate the perception of students regarding this theme. For this, a documentary research will be carried out, analysis of gaps in the curricula and a *survey* with students from Sotware Engineering course at the University of Brasília. From the information collected, recommendations will be made to address the issue of Secure Software Development in the training of students in the Software Engineering course at the University of Brasília.

**Key-words**: security, software development, training, Brazilian universities, curriculums, curricular structure, Software Engineering, survey.

# Lista de ilustrações

# Lista de tabelas

# Lista de abreviaturas e siglas

| | |
|---|---|
| ISO | International Organization for Standardization |
| DDOS | Denial of Service |
| SDL | Security Development Lifecycle |
| OWASP | Open Web Application Security Project |
| CLASP | Compreensive Lightweight Application Security Process |
| DCN | Diretrizes Curriculares Nacionais |
| NICE | National Initiative for Cybersecurity Education |
| ACM | Association for Computing Machinery |
| IEEE | Institute of Electrical and Electronics Engineers |
| SEEK | Software Engineering Education Knowledge |
| CMU | Carnegie Mellon University |
| NIST | National Institute of Standards and Technology |
| NCWF | National CyberWatch Center |
| PPC | Projeto Pedagógico do Curso |
| UnB | Universida de Brasília |
| UFC | Universidade Federal do Ceará |
| UFG | Universidade Federal de Goiás |
| UEPA | Universidade do Estado do Pará |
| UFMS | Universidade Federal de Mato Grosso do Sul |
| UFAM | Universidade Federal do Amazonas |
| UFRN | Universidade Federal do Rio Grande do Norte |
| UDESC | Universidade do Estado de Santa Catarina |
| UFTR | Universidade Federal do Paraná |

| | |
|---|---|
| UFERSA | Universidade Federal Rural do Semi-Árido |
| UNIPAMPA | Universidade Federal do Pampa |
| EUA | Estados Unidos da América |
| BLS | Bureau of Labor Statistics |
| MIT | Massachusetts Institute of Technology |
| UChicago | University of Chicago |
| JHU | Johns Hopkins University |
| UW | University of Washington |
| SWEBOK | Software Engineering Body of Knowledge |

# Sumário

# 1 Introdução

O mundo moderno tem grande dependência da utilização de sistemas de software, que se tornaram parte indispensável no funcionamento da sociedade. Exemplos são sistemas que dão suporte à vida cotidiana dos seus usuários (MCDONALD; TOWEY; BRUSIC, 2022), que apoiam o funcionamento de negócios e governos (KAZMAN; PASQUALE, 2020), que facilitam conexão entre as pessoas (SOFTWARE. . . , 2019), além de um número imensurável de outras aplicações.

A dependência que a sociedade possui de recursos de software teve destaque durante a pandemia de COVID-19 (HAGGAG et al., 2021), que evidenciou a importância destes recursos para a estrutura socioeconômica do mundo atual. São recursos que podem ser ligados à formação de opiniões a partir da difusão de informações (SOFTWARE. . . , 2019), à adoção do modelo de trabalho remoto (AKALA, 2020; BBC, 2020; KHETARPAL, 2020) ou até mesmo a sistemas essenciais nas conexões interpessoais, como as redes sociais, que fazem parte das necessidades na vida de um ser humano (HONEYCUTT; CANTRILL, 2000).

Com a presença dos serviços providos por software na maioria das funções cotidianas de cada cidadão, as informações sobre o dia-a-dia do usuário acabam virando dados importantes para o funcionamento de algumas aplicações (YASIN et al., 2019; GAñáN; CIERE; EETEN, 2017). O acesso a dados como a identificação de uma pessoa, suas ocupações ou dados sobre o seu patrimônio, passa a ser parte vital da tomada de decisões, podendo ser utilizadas por parte de negócios (JENSEN; POTTS, 2004) ou governos (MASON, 1986).

Neste contexto, informações sobre usuários passam a ter valor, não apenas para tomada de decisões, mas que também são buscadas por pessoas mal intencionadas, que utilizam diversos tipos de artifícios para praticar crimes através da internet (DEMARTINI, 2022; CETIC.BR|NIC.BR, 2022). Pesquisas demonstram aumento de até 600% na ocorrência de crimes relacionados ao uso de aplicações de software no ano de 2020 (INTERPOL, 2020), também conhecidos como *crimes cibernéticos*, que podem variar desde fraude, invasão de privacidade, até roubo e exclusão de dados (MCLEAN, 2019).

Para cada tipo de crime cibernético existe uma abordagem, uma complexidade de tratamento e prevenção, fatores que demandam uma grande diversidade de profissionais de segurança. Tais profissionais necessitam de treinamento adequado para que os sistemas de software desenvolvidos possam impedir, ou ainda prevenir tais crimes. Segundo pesquisa realizada em 2019 (CRUMPLER; LEWIS, 2022), 82% dos contratantes indicam falta de profissionais que atendam as qualificações desejadas, e 71% destes acreditam que essa

falta de profissionais gera danos às suas organizações.

Entre a necessidade de treinamento contínuo e a procura do mercado que cresce de forma desproporcional à oferta, é possível constatar que existem obstáculos impedindo que a demanda por pessoas capacitadas seja atendida. A procura por profissionais originada do contexto de segurança das empresas (INDúSTRIA, 2020), do governo (GOV.BR, 2022) e também dos usuários (DEMARTINI, 2022), que tem tomado cada vez mais conhecimento sobre falhas de segurança em aplicações, vem sendo apontada através da mídia (ANDRION, 2021).

## 1.1 O Problema

Identificada esta disparidade na oferta e na demanda, é preciso detectar os pontos de deficiência na formação de profissionais quanto às competências ligadas à segurança. A Segurança de Software e o Desenvolvimento de Software Seguro são áreas de estudo que vêm ganhando atenção mais recentemente (RASHID et al., 2019). Elas consideram que existem aspectos que não são apenas conceitos ligados à segurança de forma geral, mas sim ao processo de desenvolvimento de software, em que o desenvolvedor precisa deixar de agir de forma reativa aos problemas de segurança, passando a construir software seguro desde a sua concepção.

Apesar da importância dada a essas questões, observa-se que a introdução desses conteúdos no ensino superior tem sido lenta (ISC, 2021). Isso pode ser constatado em estruturas curriculares de casos específicos, como no curso de Engenharia de Software da Universidade de Brasília, em que a segurança de software não é tratada em disciplinas, sejam obrigatórias ou optativas, para que o discente possa desenvolver algum conhecimento sobre o desenvolvimento de uma aplicação levando em conta aspectos de segurança. Ponto que faz surgir a importante questão:

***Como melhorar o currículo dos cursos de Engenharia de Software, refletindo recomendações internacionais em relação a conteúdos de Segurança de Software e atendendo às expectativas da comunidade acadêmica quanto à formação de profissionais capazes de construir software seguro?***

## 1.2 Objetivo

O objetivo desse trabalho é produzir recomendações para promover a melhoria da formação do Engenheiro de Software em relação ao conhecimento de conceitos e práticas para o desenvolvimento de software seguro.

Os objetivos específicos são:

1. Comparar os currículos dos cursos de Engenharia de Software no Brasil e nas principais universidades internacionais quanto ao conteúdo de segurança de software.

2. Identificar lacunas na formação dos engenheiros de software no Brasil com relação à segurança de software, com base em recomendações de entidades de computação.

3. Identificar a percepção da comunidade acadêmica quanto à importância e necessidade de componentes curriculares relacionados à segurança de software.

4. Propor recomendações para melhoria dos currículos da UnB com base em modelos e padrões internacionais.

## 1.3 Metodologia

Este trabalho se caracteriza como uma pesquisa descritiva, pois se utilizará de recursos como documentos e questionários, para descrever a realidade dos cursos de Engenharia de Software e a percepção de uma amostra de indivíduos ligados ao assunto abordado.

Para a coleta de dados serão adotados os procedimentos de pesquisa bibliográfica, pesquisa documental e questionário on-line (*survey*).

A coleta de dados terá início com a análise dos currículos das universidades brasileiras, com foco nos cursos de Engenharia de Software, buscando identificar as abordagens utilizadas, ou a falta delas, para o ensino da segurança de software nas principais universidades brasileiras. Em paralelo à análise dos currículos brasileiros, será realizada também uma pesquisa das universidades consideradas como referência no área de tecnologia, com o intuito de realizar uma comparação entre estas universidades e o modelo utilizado no Brasil. Em conjunto com os dados de modelos utilizados no sistema de ensino atual, também serão analisados modelos propostos em pesquisas sobre o ensino de tópicos de segurança em currículos de graduação.

Após a realização do estudo da estrutura curricular, será aplicado um questionário (*survey*) junto aos discentes de cursos de Engenharia de Software na UnB, com o intuito de capturar a visão e o interesse dos mesmos com o desenvolvimento de conhecimentos e competências em tópicos ligados ao tema de segurança de software.

A partir resultado da etapa de análise de currículos, em conjunto com a percepção e o interesse dos atuais e futuros profissionais, serão propostas recomendações para o ensino de conteúdos ligados à segurança de software para o curso de Engenharia de Software da UnB, de forma a evidenciar a necessidade da aplicação destes conceitos durante a formação dos futuros Engenheiros de Software.

## 1.4 Organização do Documento

Este trabalho está organizado da seguinte forma:

- **Capítulo 2 - Referencial Teórico**: apresenta conceitos importantes para o entendimento do tema alvo do trabalho, fundamentando os conceitos que serão utilizados ao longo do seu desenvolvimento, com a devida pesquisa e referências bibliográficas.
- **Capítulo 3 - Metodologia**: especifica as técnicas e procedimentos metodológicos que serão utilizados para a coleta de dados, escolhas na construção dos artefatos, cuidados com a análise dos dados e também detalhes sobre a amostra de participantes na pesquisa que será realizada.
- **Capítulo 4 - Estudo dos Currículos Atuais**: Exibe os dados colhidos de universidades nacionais e internacionais, assim como a comparação e identificação das lacunas na abordagem de segurança de software no Brasil.
- **Capítulo 5 - Questionário**: Demonstra o processo de construção, divulgação e análise dos dados recolhidos com a aplicação da técnica de *survey*.
- **Capítulo 6 - Recomendações**: A partir dos dados colhidos e demonstrados nos capítulos anteriores, esta seção traz a recomendadações para a melhoria no currículo do curso de engenharia de software na UnB.
- **Capítulo 7 - Conclusão e Trabalhos Futuros**: Este capítulo sintetiza os principais achados obtidos ao longo da pesquisa, oferecendo a visão sobre o impacto das estratégias propostas para a melhoria do currículo em engenharia de software na UnB.

# 2 Segurança de Software

Quando a questão é a segurança na utilização de aplicações de software, é comum remeter-se a cenas de filmes, onde pessoas utilizam dos seus conhecimentos quase que fantasiosos sobre o funcionamento de computadores, para atacar grandes empresas ou órgãos do governo. Na realidade, a segurança de software abrange assuntos que parecem ter menos importância, como o envio de e-mails maliciosos, com o intuito de confundir o alvo, fazendo com que ele envie dados, como seu número de cartão de crédito, para o criminoso.

Em um tema tão abrangente, é importante que exista uma definição clara, tanto para o assunto abordado por este domínio, quanto para as suas áreas de atuação. Assim, é importante destacar as definições de algumas categorias que são comumente confundidas, a **segurança de software** e a **segurança da cibernética**.

- **Segurança Cibernética**: Engloba de forma geral toda a segurança de sistemas da informação, sua estrutura lógica(software) e física(hardware), assim como a infra-estrutura associada, dados armazenados, serviços fornecidos, acessos indevidos e os prejuízos gerador pela deficiência na segurança (OFFICE et al., 2017).

- **Segurança de Software**: Se preocupa em satisfazer os objetivos específicos ou implícitos de segurança de um sistema de software. Esses objetivos estão relacionados à confidencialidade, integridade e disponibilidade dos dados e funcionalidade do sistema (RASHID et al., 2019).

A partir das definições, é possível perceber que a segurança de software é uma subcategoria da segurança cibernética. Enquanto a segurança cibernética trata de todas as estruturas que podem afetar o funcionamento de um sistema, a segurança de software busca apenas garantir a **confidencialidade**, **disponibilidade** e **integridade** do software (ISO/IEC, 2018). A ISO 27000 identifica tais propriedades da segurança de software como:

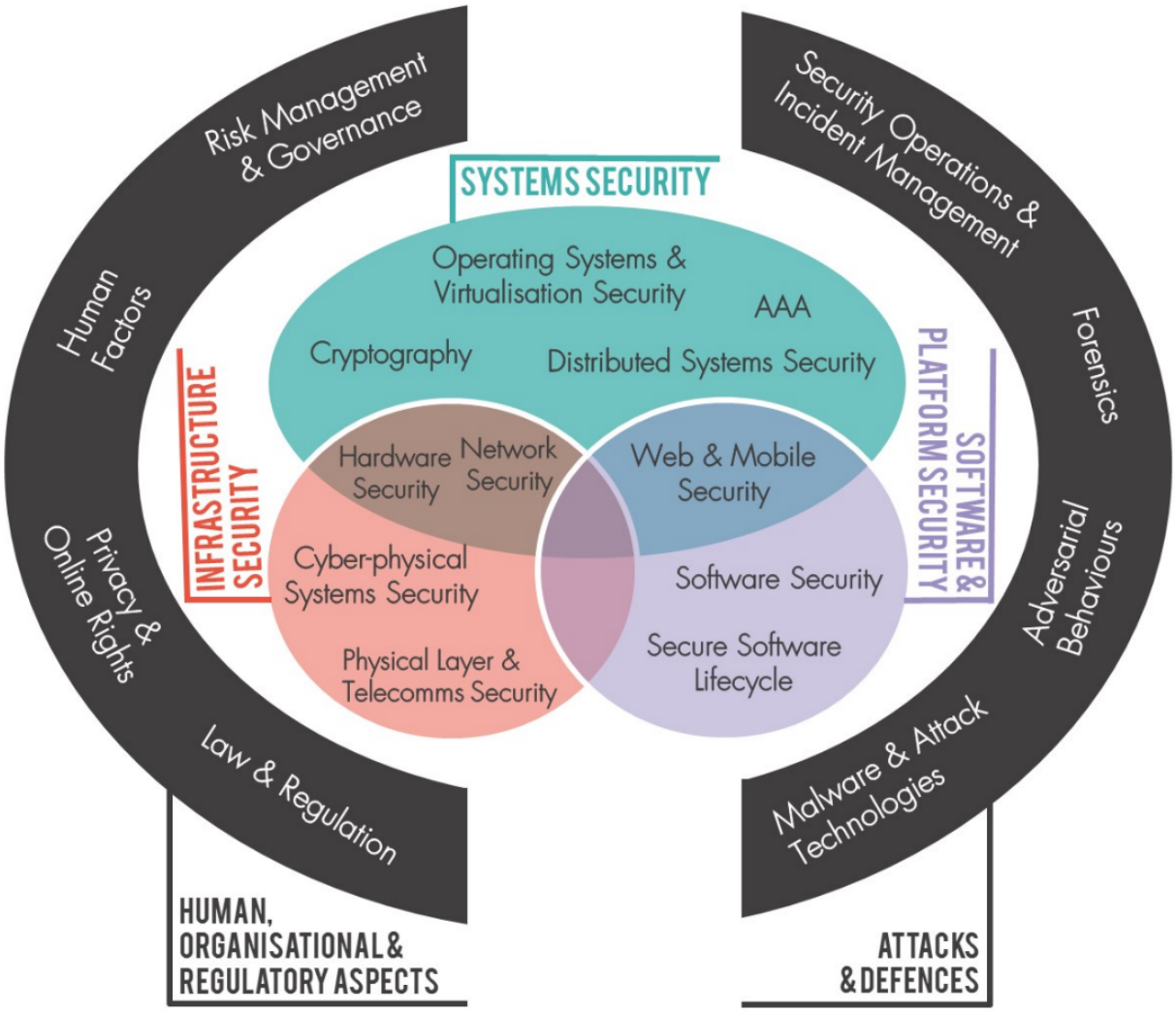
- **Confidencialidade**: Se refere a propriedade que a funcionalidade não deverá estar acessível a pessoas, organizações ou processos não autorizados.

- **Disponibilidade**: Se refere a propriedade de estar acessível e utilizável sob a demanda de uma entidade autorizada.

- **Integridade**: Se refere a propriedade de funcionamento preciso e pleno do sistema.

A Figura 1 ilustra o escopo da segurança cibernética, onde é possível observar que a segurança de software faz parte da subcategoria de **Software & Plataform Security**, representada pelo circulo roxo.

Figura 1 – Escopo da segurança cibernética




Fonte: (RASHID et al., 2019)

A segurança de software precisa lidar com uma grande variedade de vulnerabilidades de segurança. Entre os diversos tipos de vulnerabilidades conhecidas, é possível citar ataques de DDoS(*Denial of Service*), falsificação digital, roubo de dados, *malwares*, violação de dados e sistemas, falhas de sistema como *buffer overflow*, entre muitas outras.

É possível perceber que a segurança de software, mesmo já tratando de uma parte mais específica de problemas quando comparada à segurança cibernética, ainda precisa tratar de inúmeras fragilidades. Vulnerabilidades como as citadas são bem conhecidas na área de segurança de software atual, e ainda assim são preocupações constantes dos profissionais. Porém, as preocupações não se esgotam apenas nas falhas conhecidas, já que também é preciso se preocupar com as novas vulnerabilidades, que são descobertas todos os dias (IWENDI et al., 2020).

Com a recorrência de vulnerabilidades antigas e bem conhecidas e o advento de novas metodologias de ataque, **quais providências** devem ser tomadas para que um software seja seguro e **como incorporar** essas medidas de proteção na aplicação, são dois tópicos fundamentais para a construção de um software seguro.

As providências tomadas por desenvolvedores tem historicamente dado foco ao fortalecimento da segurança na camada de rede da aplicação, com a utilização de *firewalls* (RASHID et al., 2019), sendo que cerca de 70% das vulnerabilidades de segurança são encontradas na camada de aplicação (FUTCHER; SOLMS, 2008).

Já o modo de aplicação das medidas de segurança em um software tem em geral ocorrido de forma reacional, onde a segurança só é implementada no produto na ocasião em que se encontra uma vulnerabilidade. De acordo com Rashid et al. (2019), isso tende a gerar diversos problemas, tais como:

- **Invasores podem se aproveitar das vulnerabilidades sem serem descobertos**: os invasores podem se aproveitar da brecha por um período extenso de tempo, antes que ela seja encontrada e sanada.
- **Brechas de segurança custam caro**: além de milhões de dólares gastos anualmente com as brechas de segurança, as empresas ainda precisam lidar com o dano que isso causa à sua reputação.
- **Correções introduzem novas vulnerabilidades**: Com o caráter reacional das correções e a urgência ligada à situação, as correções são feitas às pressas e muitas vezes introduzem novas brechas.
- **Correções são comumente ignoradas pelos usuários**: Usuários podem se mostrar contrários à aplicação das correções de segurança.

Levando estes pontos em consideração, em 2002 surge a primeira bibliografia (RASHID et al., 2019) com a proposta de um método de desenvolvimento que cuida da segurança da aplicação desde a sua concepção, o **desenvolvimento seguro de software** (VIEGA; MCGRAW, 2002).

## 2.1 Desenvolvimento Seguro de Software

O desenvolvimento seguro de software procura introduzir a segurança como parte de todas as etapas de criação de um produto. Tais etapas de desenvolvimento não necessariamente envolvem apenas a escrita de código, mas também compreendem desde o treinamento de profissionais, até a utilização de componentes de terceiros.

O assunto de uma abordagem que trata da segurança de software além da metodologia reacional de aplicação de segurança, surgiu no ano de 1998 com McGraw (1998), fruto da sua pesquisa realizada com o intuito de investigar a aplicação de software na avaliação de vulnerabilidades de outras aplicações. No ano de 2002 a área ganhou o seu primeiro livro, onde Viega e Mcgraw (2002) focam na integração da segurança no processo desenvolvimento de software.

As metodologias de desenvolvimento seguro estavam surgindo no início dos anos 2000 e tiveram destaque com um incidente envolvendo a exposição de falhas de segurança em um dos produtos da empresa Microsoft. O acontecimento demonstrou o impacto que a exposição de uma vulnerabilidade poderia custar a uma empresa (RASHID et al., 2019). Em consequência deste incidente, a importância da segurança de software ganhou destaque, principalmente dentro da Microsoft, de tal maneira que o diretor executivo na época, Bill Gates, enviou um memorando para toda a empresa (GATES, 2002), descrevendo um novo foco de desenvolvimento dentro da empresa daquele momento em diante.

Entre os assuntos abordados pelo memorando, é possível destacar algumas seções importantes (GATES, 2002):

- "... Hoje, no mundo desenvolvido, não nos preocupamos com a disponibilidade dos serviços de água e eletricidade."
- "... Com a telefonia, nós contamos com a disponibilidade e a segurança para conduzir transações de negócio altamente confidenciais, sem se preocupar com as informações que estão sendo transmitidas."
- "Eventualmente, nosso software deve ser tão fundamentalmente confiável que os consumidores nunca precisarão se preocupar com isso."

Não apenas nestes pontos, mas no decorrer de todo o memorando, Gates tenta definir o que seria a *Trustworthy Computing* (ou computação digna de confiança), trazendo à tona noções consideradas essenciais para a segurança de software, a **disponibilidade**, a **confiabilidade** e a **integridade** dos sistemas. A partir disso, o processo de desenvolvimento dentro da Microsoft foi reestruturado, trazendo a segurança dos produtos ao foco durante a sua construção, tornando a empresa uma referência em segurança de software (RASHID et al., 2019).

A partir das novas práticas adotadas pelo time de desenvolvimento dentro da Microsoft, a metodologia de desenvolvimento utilizada pela empresa ganhou fundamento e acabou sendo publicada como um guia de práticas (HOWARD; LEBLANC, 2003), que deveriam ser seguidas para a construção de software seguro. Dentro deste processo de desenvolvimento seguro, um destaque importante está no preparo dos indivíduos que vão

participar do processo, uma vez que a leitura do livro publicado com as práticas utilizadas dentro da empresa era um requisito para todos os desenvolvedores. Atualmente, a metodologia *Security Development Lifecycle* (**SDL**) se baseia em 12 etapas (MICROSOFT, 2022):

1. **Fornecer treinamento**: Segurança é o trabalho de todos"(MICROSOFT, 2022). Aqui, o método trata todos os envolvidos no projeto como indivíduos que necessitam de conhecimento de segurança de software, não importando sua área de atuação. Partindo disso, os envolvidos são instruídos não apenas nas técnicas e conceitos de segurança, mas também a pensar como o indivíduo gerador de ataques.

2. **Definir os requisitos de segurança**: Diz respeito ao ato de introduzir a definição de requisitos de segurança durante o design inicial do projeto, durante a sua concepção, para que os estágios de planejamento possam integrar a segurança de forma a causar menor distúrbio no desenvolvimento dos requisitos funcionais do sistema.

3. **Definir métricas e relatórios de conformidade**: Trata da importância da definição de um padrão a ser seguido durante o projeto. Padrão que deve servir de guia para o desenvolvimento, além dos efeitos que isso gera no trabalho de um time.

4. **Modelagem de ameaças**: Aplicado em sistemas que possuem risco significativo de segurança, a modelagem de ameças pode ser aplicada em diversos níveis do sistema. Pratica que permite que o time possa considerar, discutir e documentar essas possíveis ameaças.

5. **Estabelecer os requisitos de design**: Tratando da implementação de funcionalidades seguras, e o importante ponto que em grande parte dos casos, a implementação de uma funcionalidade acaba gerando novos problemas de segurança, reforçando o ponto de que a segurança deve ser implementada de forma consistente durante o decorrer do projeto.

6. **Definir e usar padrões de criptografia**: Pratica que leva em conta o impacto da adoção de padrões de criptografia. Padrões responsável por garantir a integridade e confidencialidade de dados transmitidos ou armazenados. Levantando o ponto de que é importante considerar alternativas que podem ser facilmente alteradas se necessário.

7. **Gerenciar riscos de segurança provenientes da utilização de componentes de terceiros**: Pratica que trata da importância de manter um catálogo das dependências de terceiros que são utilizadas dentro do projeto, todas com respectivos planos para mitigar eventuais vulnerabilidades de segurança dentro destes pacotes.

8. **Utilizar ferramentas aprovadas**: Definir e publicar uma lista de ferramentas e as suas utilidades com foco na segurança do projeto. Aqui também é levantada o incentivo de se utilizar as versões mais atuais de tais ferramentas, visando novas análises e funcionalidades.

9. **Executar analise estática de testes de segurança**: Sendo normalmente implementado em esteiras de *Continuous Integration* (CI) ou em ambiente de desenvolvimento, a *Static Analysis Security Testing* (SAST) busca acrescentar um método de revisão de código, garantindo que as políticas de código estão sendo seguidas.

10. **Executar análise dinâmica de testes de segurança**: Assim como a SAST, a *Dynamic Analysis Security Testing* (DAST) também é um tipo ferramenta de testes de segurança. Diferentemente da SAST, que analisa o código antes da compilação, a DAST foca em realizar análises em tempo de execução.

11. **Executar testes de penetração**: Atividade realizada por profissionais experientes, com vasto conhecimento no funcionamento da segurança. As ações realizadas nesta prática visam simular ataques de hackers, buscando descobrir potenciais vulnerabilidades que podem ser originadas de erros de código, sistema, configuração e etc.

12. **Estabelecer um padrão para o processo de resposta à incidentes**: Em casos onde as medidas tomadas para a prevenção de vulnerabilidades não é o necessário para evita-las, esta prática visa criar um plano de ação para ajudar a combater as novas vulnerabilidades que são descobertas.

Vale destacar a primeira prática, **fornecer treinamento**, a qual reitera que a segurança é responsabilidade de todos, não apenas dos envolvidos diretamente no processo de desenvolvimento. Assim, todos devem ter conhecimento sobre a segurança, conhecimento que não necessariamente deve ser em nível de um especialista, mas o entendimento do contexto de segurança, assim como o ponto de vista do indivíduo que está realizando o ataque, é parte fundamental para qualquer pessoa que faz parte do projeto.

Além do ciclo de vida para desenvolvimento seguro da Microsoft, denominado de SDL (MICROSOFT, 2023b), existem outras metodologias para o desenvolvimento seguro de software, que propõe variações em relação às orientações e processos seguidos, como a *Compreensive Lightweight Application Security Process* (**CLASP**) da *Open Web Application Security Project* (OWASP)(OWASP, 2006) e também a metodologia **touchpoints** (MCGRAW, 2006).

## 2.2 Segurança de Software na Educação

Mesmo com modelos bem estruturados e testados de segurança de software, apenas uma pequena parcela dos desenvolvedores chegam ao mercado de trabalho com capacidade de aplicá-los aos projetos. A deficiência na abordagem do assunto de segurança pode ser atribuída a diversos fatores, mas talvez o que mais se destaca é a falta de treinamento dos educadores em práticas de desenvolvimento seguro, já que este assunto também não foi abordado em sua formação (ZHU; LIPFORD; CHU, 2013). Adicionalmente, observa-se que as atuais Diretrizes Curriculares Nacionais (DCNs) para os cursos da área de Computação não mencionam a segurança, de forma direta, na lista de competências de um egresso do curso de Engenharia de Software (MEC, 2016).

Uma vez que foi identificada a carência de treinamento de profissionais, a pesquisa e busca por incentivar medidas que abordassem o problema recaiu sobre instituições governamentais. Entre essas iniciativas é possível citar a *National Security Agency* (NSA), com o desenvolvimento de diversos módulos para segurança (LODGHER; YANG, 2017), a *National Initiative for Cybersecurity Education* (NICE), com a intenção de aumentar o número de profissionais de segurança disponíveis no mercado (NEWHOUSE et al., 2017), a Estratégia Nacional de Segurança Cibernética de 2016-2021, proposta pelo governo do Reino Unido com a intenção de promover pesquisa e formação de profissionais (YANG; LODGHER; LEE, 2018), entre outras iniciativas.

Baseado nos recursos que uma universidade possui, Yuan e Yang (2016) destacam três abordagens para a implementação de segurança no currículo de Engenharia de Software:

- **Segurança como um curso isolado**: em que a segurança de software deveria ser tratada de forma mais aprofundada, formando profissionais com conhecimento pleno sobre a segurança de software.

- **Segurança como um módulo ou disciplina do curso**: oferta de disciplinas obrigatórias ou opcionais em cursos ligados à construção de software, que tratam dos conceitos de segurança no desenvolvimento de aplicações.

- **Segurança de forma difusa**: abordagem da segurança como um ponto de importância durante toda a formação dos desenvolvedores, dentro de várias disciplinas, assim como a eficiência, complexidade e estruturação de código.

Nos artigos analisados, é possível notar uma tendência na sugestão de dispersar a implementação do conteúdo ao longo do curso. No entanto, também são citados problemas nesta abordagem:

- "... No entanto, encontrar maneiras de adicionar o conteúdo de segurança em conjunto com uma multitude de outros temas com evolução da Ciência da computação pode ser uma tarefa desafiadora" (BLAIR et al., 2020);
- "Pode ser que não haja tempo hábil para cobrir os tópicos ensinados em disciplinas, se houver a adição de tópicos de segurança" (YUAN; YANG, 2016);
- "É difícil treinar um grande número de professores em Engenharia de Software Seguro" (YUAN; YANG, 2016);
- "Não é realista a expectativa de que a faculdade que ensina o tópico esteja atualizada com as vulnerabilidades mais recentes" (YUAN; YANG, 2016).

Assim, é importante notar que qualquer uma das alternativas para a implementação do conteúdo no curso tem seus desafios e estes precisam ser bem analisados para que se possa ter maior aproveitamento da abordagem.

## 2.3 Abordagens para o Ensino de Segurança de Software

Para que se pudesse tratar dos problemas levantados na implementação de módulos de segurança durante a formação dos profissionais, foram realizados diversos estudos e trabalhos voltados para propor alternativas voltadas para o ensino da segurança de software.

O ensino de segurança abrange diversos níveis de formação, abordando iniciativas com o intuito de servir como referência para o ensino de segurança (NEWHOUSE et al., 2017), para mestrado ou bacharelado em cursos como Ciência da Computação (BLAIR et al., 2020) ou Engenharia (JAHN; MOTTOK, 2020), até mesmo módulos que podem ser introduzidos durante o ensino médio (LODGHER; YANG; BULUT, 2018). Para propósito deste trabalho, serão apresentados os trabalhos ligados ao curso de Engenharia de Software.

### 2.3.1 ACM/IEEE - SE 2014

O ACM/IEEE - SE 2014 (IEE; ACM, 2015) é um currículo desenvolvido pelo empenho do grupo focado em educação da *Association of Computing Machinery* (ACM) em conjunto com o grupo atividades voltadas para a computação do *Institute of Electrical and Electronics Engineers* (IEEE), com foco em criar recomendações para disciplinas voltadas para a computação. O documento resultado deste trabalho conjunto (IEE; ACM, 2015), tem o intuito de servir como guia sobre o que deve estar presente no currículo de um bacharel em Engenharia de Software.

Esta referência destaca a segurança como uma área de conhecimento apropriada para um bacharel de Engenharia de Software. Esse conhecimento é designado como *Software Engineering Education Knowledge* (SEEK) e engloba 10 áreas de conhecimento, sendo elas:

- Fundamentos de computação(CMP);
- Fundamentos de matemática e engenharia(FND);
- Prática profissional(PRF);
- Modelagem e análise de software(MAA);
- Análise e especificação de Requisitos(REQ);
- Design de software(DES);
- Verificação e validação de software(VAV);
- Processo de software(PRO);
- Qualidade de Software(QUA);
- **Segurança(SEC)**.

A segurança é tratada como dois componentes distintos e relacionados. Um deles trata da proteção de informação, sistemas e rede. O outro leva ao foco a incorporação da segurança durante todo o processo de desenvolvimento de um produto de software.

A abordagem proposta segue a seguinte estrutura apresentada na Figura 2.

A Figura 2 divide o tema em Fundamentos da Segurança (*Security Fundamentals*), Segurança de Computador e Rede (*Computer and network security*) e Desenvolvendo Software Seguro (*Developing secure software*). Assim como as divisões e os assuntos abordados em cada uma delas, a tabela também especifica qual habilidade cognitiva cada assunto aborda:

- **Knowledge (k)**: Relembrar e apresentar informações na forma de materiais.
- **Comprehension (c)**: Entender a informação e o significado do material apresentado.
- **Application (a)**: Utilizar os conceitos aprendidos em situações concretas.

A tabela apresenta também a relevância dos assuntos para o núcleo do conhecimento da Engenharia de Software:

Figura 2 – Estrutura de referência para abordagem de segurança

| Reference | | k,c,a | E,D | Hours |
|---|---|---|---|---|
| SEC | **Security** | | | 20 |
| SEC.sfd | Security fundamentals | | | 4 |
| SEC.sfd.1 | Information assurance concepts (confidentiality, integrity, and availability) | k | E | |
| SEC.sfd.2 | Nature of threats (e.g., natural, intentional, and accidental) | k | E | |
| SEC.sfd.3 | Encryption, digital signatures, message authentication, and hash functions | c | E | |
| SEC.sfd.4 | Common cryptographic protocols (applications, strengths, and weaknesses) | c | E | |
| SEC.sfd.5 | Nontechnical security issues (e.g., social engineering) | c | E | |
| SEC.net | Computer and network security | | | 8 |
| SEC.net.1 | Network security threats and attacks | k | E | |
| SEC.net.2 | Use of cryptography for network security | k | E | |
| SEC.net.3 | Protection and defense mechanisms and tools | c | E | |
| SEC.dev | Developing secure software | | | 8 |
| SEC.dev.1 | Building security into the software development life cycle | c | E | |
| SEC.dev.2 | Security in requirements analysis and specification | a | E | |
| SEC.dev.3 | Secure design principles and patterns | a | E | |
| SEC.dev.4 | Secure software construction techniques | a | E | |
| SEC.dev.5 | Security-related verification and validation | a | E | |

Fonte: (IEE; ACM, 2015)

- **Essential (E)**: O tópico faz parte do núcleo.
- **Desirable (D)**: Tópico não faz parte do núcleo, mas deveria ser incluído se possível; ou então é tratado como material optativo.

Por fim, é especificado o tempo em horas considerado mínimo para a abordagem das seções mencionadas.

### 2.3.2 Módulos de Segurança para Cursos de Graduação em Engenharia de Software

Yang, Lodgher e Lee (2018) discutem a importância da educação em Engenharia de Software Segura e apresentam uma proposta para uma área de conhecimento, currículo e recursos para a incorporação de princípios de segurança em programas de ensino de Engenharia de Software. Os autores argumentam que a segurança deve ser um componente fundamental do ensino e que os desenvolvedores de software devem ser treinados para criar sistemas seguros desde o início do desenvolvimento. A proposta inclui uma estrutura para a incorporação de segurança em vários níveis do currículo de Engenharia de Software, bem como recursos adicionais para ajudar os professores a ensinar Segurança de Software.

Tendo como princípio que a maioria das vulnerabilidades de sistemas da informação são causados por defeitos de software, o artigo traz uma análise de estruturas curriculares que já existem, como no modelo utilizado para o ensino na *Carnegie Mellon University* (CMU) e estruturas curriculares propostas em recursos como o *Software Assurance Common Body of Knowledge*, proposto pela NIST.

Com o intuito de introduzir o conceito de técnicas de desenvolvimento de segurança de forma sistemática ao longo do processo de construção de uma aplicação, o artigo sugere a introdução de nove módulos no currículo de formação de um bacharel em Engenharia de Software. Tais módulos tratam de uma série de tópicos voltados para a instalação de requisitos robustos de segurança, design de software seguro e verificação de software seguro através do ciclo de desenvolvimento de software.

A Figura 3 apresenta os módulos propostos pelo estudo, indicando os objetivos e resultados de aprendizagem dos estudantes, as horas necessárias para que um estudante complete os módulos nos cursos de Engenharia de Software e as categorias de currículo proposto em estudo realizado pela instituição NICE (NEWHOUSE et al., 2017).

Figura 3 – Módulos de segurança propostos

| *Module No.* | *Student Learning Outcomes / Objectives* | *Hours* | *Course(s)* | |
|---|---|---|---|---|
| 1 | Describe the requirements for integrating security into the software development lifecycle | 4 | ***Secure Software Requirement Specification*** | SW I |
| 2 | Apply the risk management on a software project plan | 4 | | SW I |
| 3 | Specify functional requirements and identifies the expected execution paths. | 5 | | SW I |
| 4 | Apply the concepts of the Design Principles for Protection Mechanisms, the Principles for Software Security, and the Principles for Secure Design on a software development project | 7 | ***Secure Coding Practice*** | SW I |
| 5 | Describe software development best practices for minimizing risks and vulnerabilities in software development. | 7 | | SW I/II |
| 6 | Identify security test cases of a software application | 4 | ***Security Testing and Verification*** | SW II |
| 7 | Conduct a software security testing | 4 | | SW II |
| 8 | Conduct a security verification and assessment of a software application | 5 | | SW II |
| 9 | Use reverse engineering tools to safely perform static and dynamic analysis of software (or malware) of potentially unknown origin | 5 | ***SW Reverse Engineering*** | SW II |

Fonte: (YANG; LODGHER; LEE, 2018)

# 3 Metodologia

Este capítulo descreve o plano metodológico utilizado durante o desenvolvimento da pesquisa realizada. Para isto, a metodologia foi classificada quando a sua natureza, abordagem e procedimentos técnicos.

A natureza é identificada como aplicada, tendo em vista que este trabalho se propõe a gerar recomendações para lidar com um problema específico. Quanto à abordagem, o trabalho foca na utilização da pesquisa qualitativa, com o intenção de compreender e interpretar os fenômenos complexos e subjetivos ligados ao tema em questão, se utilizando da coleta de dados não-numéricos (MINAYO, 2012). Quanto aos objetivos, o trabalho se caracteriza como uma pesquisa descritiva, pois se utiliza de recursos como pesquisa bibliográfica, pesquisa documental e questionário para descrever a realidade e a percepção de uma amostra de indivíduos.

Para a execução da pesquisa, foram utilizados os seguintes procedimentos técnicos:

- **Pesquisa Bibliográfica**: um método de investigação, que se utiliza materiais já elaborados como livros e artigos (GIL, 2008). Utilizada na forma de consultas a pesquisas e estudos com o enfoque em educação em segurança de software A pesquisa bibliográfica foi aplicada para evidenciar os pontos considerados importantes para a formação de profissionais capazes de produzir software que trata a segurança com a devida relevância, quando levado em conta os padrões exigidos pelo mercado.

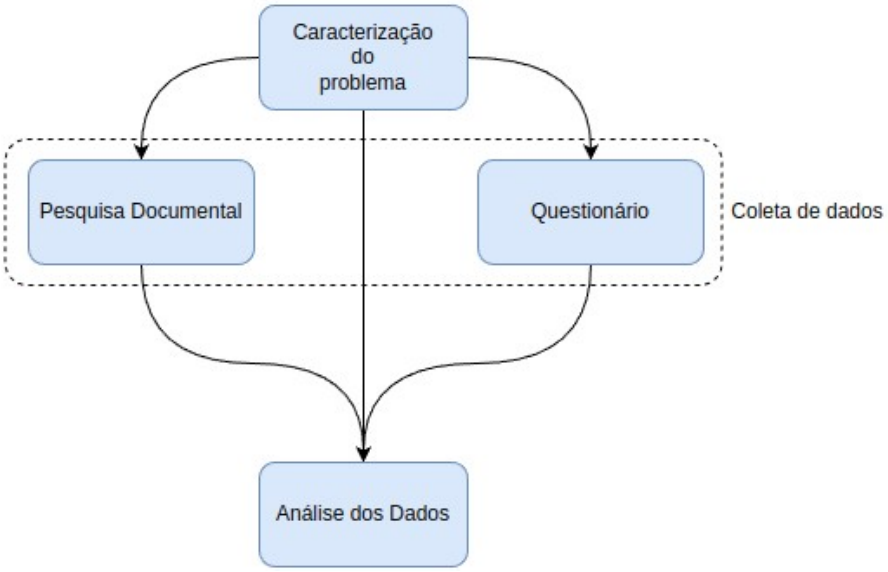
- **Pesquisa Documental**: diferente da pesquisa bibliográfica, a pesquisa documental busca realizar uma análise diretamente em arquivos que ainda não foram analisados (GIL, 2008). Este procedimento foi utilizado na forma de pesquisa e análise de estruturas curriculares. Em casos nacionais foram analisados os documentos de Projeto Pedagógico do Curso (PPC) e em casos internacionais foram analisadas as estruturas curriculares de cursos de Ciência da Computação, ambos com a intenção de evidenciar os modelos utilizados para a abordagem do tema de segurança na formação de engenheiros de software.

- **Pesquisa(*survey*)**: tem o propósito de construir modelos explanatórios a partir da análise de uma determinada população (WOHLIN et al., 2012), ou para validar conhecimento (PFLEEGER; KITCHENHAM, 2001). Isso se dá a partir de perguntas, que são aplicadas no seu instrumento de *questionário*, normalmente em uma parcela representativa do público escolhido (JR, 2013). O questionário foi utilizado neste trabalho com o intuito de mensurar a opinião de uma amostra do público escolhido para do estudo (atuais e futuros engenheiros de software). Neste sentido, buscamos

captar a perspectiva dos participantes sobre a abordagem atual da segurança em cursos formadores, assim como a importância deste tópico quando comparado a outras capacidades desenvolvidas durante a formação.

O plano metodológico proposto para este trabalho foi dividido em três etapas, como demonstrado na Figura 4, etapas que serão discutidas nas próximas seções.

Figura 4 – Etapas do plano metodológico adotado neste trabalho

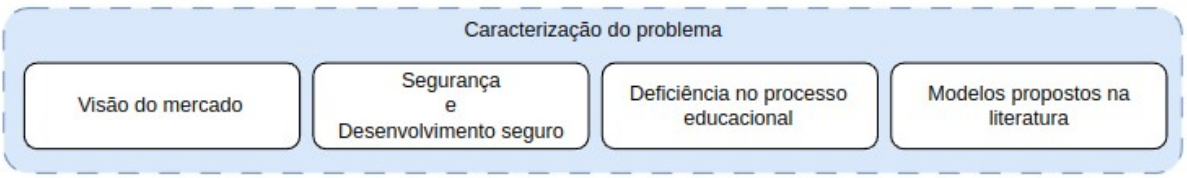



## 3.1 Caracterização do Problema

Utilizando-se dos procedimentos de pesquisa documental e bibliográfica, esta etapa buscou contextualizar o assunto alvo da pesquisa, assim como demonstrar qual a visão do mercado de desenvolvimento de software sobre a segurança. Também foi objetivo desta fase, evidenciar a falta de profissionais capacitados para a produção de software seguro, com conhecimentos necessários de identificação e prevenção de falhas relacionadas a segurança de código.

Como demonstrado na Figura 5, esta etapa foi subdividida em quatro tarefas:

Figura 5 – Metodologia - Atividades da etapa de caracterização do problema



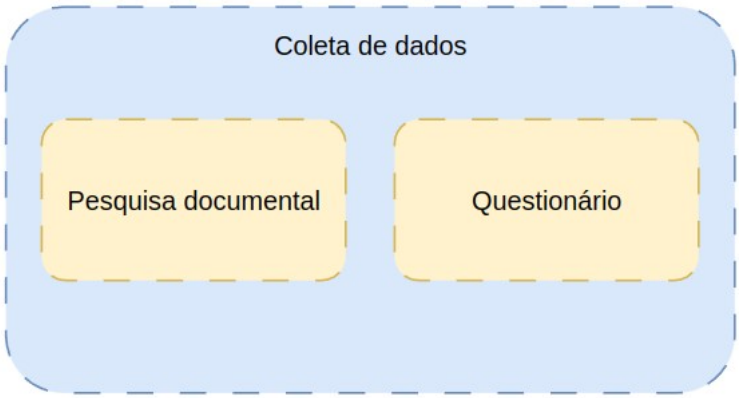
- **Visão do mercado**: demonstrar a procura do mercado por profissionais capazes de construir aplicações seguras.
- **Segurança e desenvolvimento seguro**: descrever as particularidades da segurança e o desenvolvimento seguro.
- **Deficiência no processo educacional**: expor as lacunas encontradas nos modelos de ensino atuais, mais especificamente em currículos da formação de engenheiros de software brasileiros.
- **Modelos propostos na literatura**: apresentar modelos propostos por estudos relacionados ao ensino de segurança.

## 3.2 Coleta de dados

Esta etapa buscou reunir informações sobre o ensino do segurança de software e sobre modelos de currículos considerados como referência na área. Adicionalmente, esta etapa buscou coletar as opiniões da comunidade de desenvolvedores em formação (discentes) e dos encarregados em promover o ensino (docentes) através de um questionário.

A Figura 6 ilustra as atividades realizadas nesta etapa, que está subdividida em duas seções que ocorrem simultaneamente: a pesquisa documental e o questionário.

Figura 6 – Metodologia - Atividades da etapa de coleta de dados



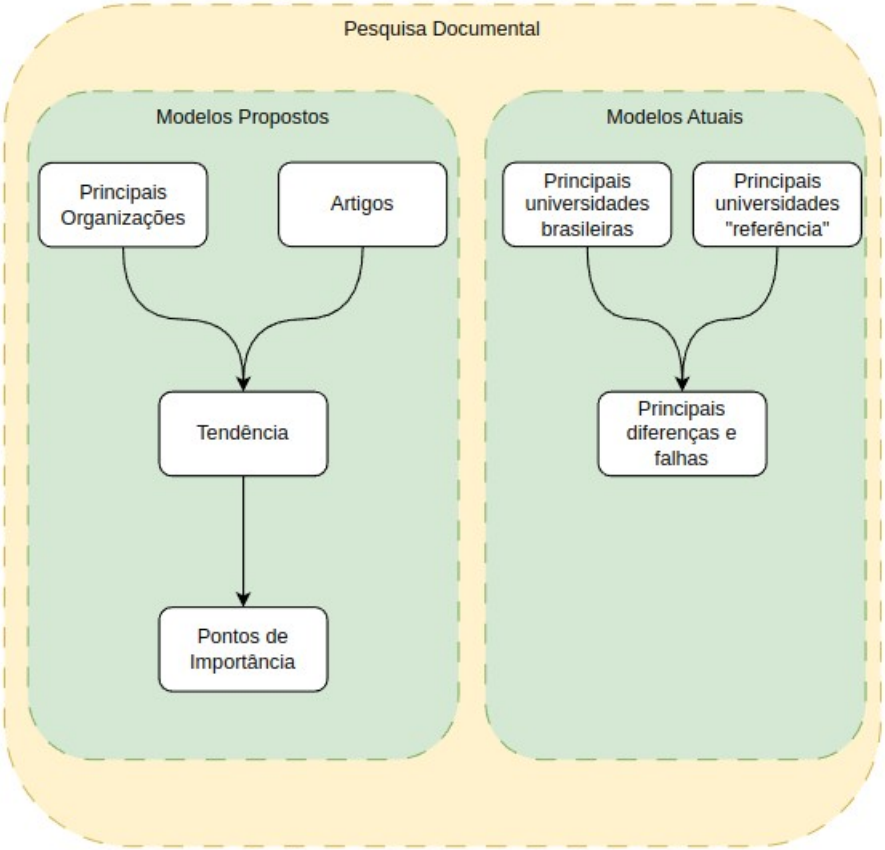

### 3.2.1 Pesquisa Documental

A seção de **Pesquisa Documental** é dedicada à busca de informações existentes sobre os métodos e estruturas utilizados para a formação dos profissionais da área de desenvolvimento de software. Esta busca foi realizada com foco em modelos utilizados atualmente e modelos considerados como referência na área.

Esta atividade foi realizada através da pesquisa bibliográfica relacionada ao período de 2012-2022 em conjunto com a pesquisa documental dos currículos vigentes em universidades nacionais e internacionais.

Para isso, como demonstra a Figura 7, esta seção foi subdividida em duas categorias:

Figura 7 – Coleta de dados - Atividades da pesquisa documental



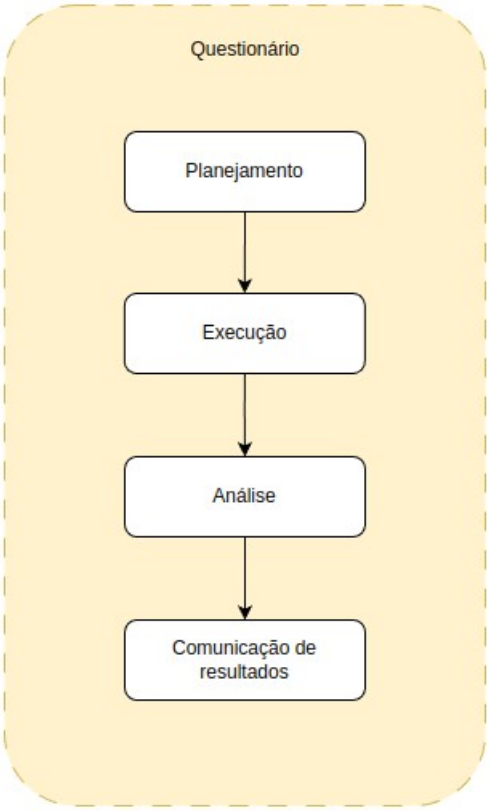

- **Modelos propostos**: busca reunir trabalhos de importância, publicados, como propostas para a implementação de módulos de segurança em currículos de formação acadêmica.
- **Modelos atuais**: busca identificar os currículos utilizados atualmente no contexto nacional, bem como currículos internacionais considerados como "referência" em relação a área de desenvolvimento de software.

### 3.2.2 Questionário

Utilizando-se do método de **Pesquisa**, esta seção tem como objetivo coletar dados relacionados à percepção dos atuais e futuros engenheiros de software sobre a importância da abordagem de conceitos sobre segurança durante a sua formação.

A Figura 8 apresenta uma divisão das etapas executadas durante a aplicação da metodologia de pesquisa, seguindo o modelo proposto por Molléri, Petersen e Mendes (2019), para estruturar o processo.

Figura 8 – Coleta de dados - Atividades para aplicação do questionário




### 3.2.3 Planejamento

Seção onde ocorre a definição do instrumento, desde a caracterização dos objetivos até a validação do instrumento de questionário produzido. Esta seção é composta pelas seguintes etapas:

- **Objetivos de pesquisa**: com a intenção de especificar corretamente o objetivo do questionário. Neste ponto, o questionário é direcionado para transparecer a visão de engenheiros de software sobre a característica de segurança, como membro do time desenvolvedor, usuário de aplicações e também durante a sua formação, assim como os tópicos importantes para uma disciplina com foco em segurança.

- **Plano de estudo**: Estudo e registro dos procedimentos utilizados durante o processo de construção da pesquisa e, normalmente, das perguntas presentes no questionário. Tendo como base o artigo Zatko (2016) para escolha do tema e abordagem, em conjunto com o *checklist* proposto em Molléri, Petersen e Mendes (2019), foi

possível traçar um plano de ação, quer é melhor descrito no capítulo voltado ao *Questionário*.

- **Identificar a população**: definir a amostra da população que será alvo do questionário, escolha que impacta diretamente o plano de amostragem. Seguindo a definição dos objetivos, a pesquisa será direcionada com foco em discentes de Engenharia de Software da UnB.
- **Plano de amostragem**: Fase onde são definidos os planos para a escolha da amostra representativa da população. A forma de divulgação escolhida foi de divulgar o questionário em grupos de desenvolvedores, alunos e professores da Engenharia de Software da UnB.
- **Projetar o instrumento**: etapa onde são definidas a versão preliminar do instrumento, características da aplicação e forma de administração. Aqui, a proposta é de administrar um questionário que será feito remotamente pelo participante, com questionários produzidos utilizando a ferramenta online Microsoft forms.
- **Validação do instrumento**: Etapa em que o instrumento passa por uma fase de testes. Onde é mensurada a capacidade do instrumento de medir o que é alvo da pesquisa, assim como a revisão de especialistas ou participantes comuns. Neste ponto, a proposta é de aplicar o questionário inicial em um pequeno grupo, com três ou quatro pessoas, visando principalmente observar o tempo de resposta e possíveis dificuldades dos participantes.

### 3.2.4 Execução

Com o instrumento da pesquisa devidamente construído e validado, se inicia a etapa de execução, onde a aplicação do método se divide em duas etapas:

- **Recrutamento de participantes**: com o método utilizado devidamente selecionado na etapa de **Plano de amostragem**, esta etapa adota medidas que podem ter auxílio de ferramentas, para investigar possíveis problemas no engajamento dos participantes. Visando obter o maior número de respostas, detalhes como duração média do questionário e possíveis incentivos serão analisados.
- **Gestão de respostas**: as medidas que serão tomadas após a distribuição do instrumento, com a intenção de garantir o número adequado de respostas.

### 3.2.5 Análise dos dados

Após a coleta, é importante que os dados sejam validados, buscando remover dados inválidos ou incompletos. A seguir os dados serão organizados usando tabelas,

gráficos e estatística descritiva de forma a apresentar de forma sumarizada a percepção dos participantes a respeito da importância da abordagem do tema de segurança na formação do engenheiro de software.

### 3.2.6 Produto final

Este trabalho se propôs a produzir recomendações para abordagem da segurança de software no curso de Engenharia de Software da UnB.

A partir das análises realizadas tanto na pesquisa documental, quanto na pesquisa bibliográfica, foi possível identificar os pontos que são importantes para o desenvolvimento de habilidades relacionadas a segurança. Baseado nestes pontos, foram recomendados novos componentes curriculares ou conteúdos a serem adicionados aos componentes existentes, levando em conta os impactos que a abordagem de determinados temas trazem ao curso, principalmente quanto à sua complexidade e à sobrecarga de conteúdo.

Uma vez que a proposta de ajuste do currículo foi formada a partir da coleta de dados e da análise de documentos e estudos, o resultado do questionário serviu para avaliar o nível de conscientização dos discentes a respeito da demanda pelo assunto dentro do ambiente de formação de profissionais.

# 4 Estudo dos currículos atuais

Para entender melhor quais abordagens são utilizadas atualmente para a formação de engenheiros de software, quanto ao tema da segurança, foi realizado um levantamento de dados. A busca foi composta por currículos utilizados no âmbito nacional, a partir do conjunto de universidades públicas que oferecem o curso de Engenharia de Software e em âmbito internacional, a partir de uma análise realizada utilizando universidades com melhor ranking segundo avaliações mundiais (EDURANK, 2021; EDUCATION, 2023; USNEWS, 2023) para a formação de engenheiros de software.

## 4.1 Coleta de dados

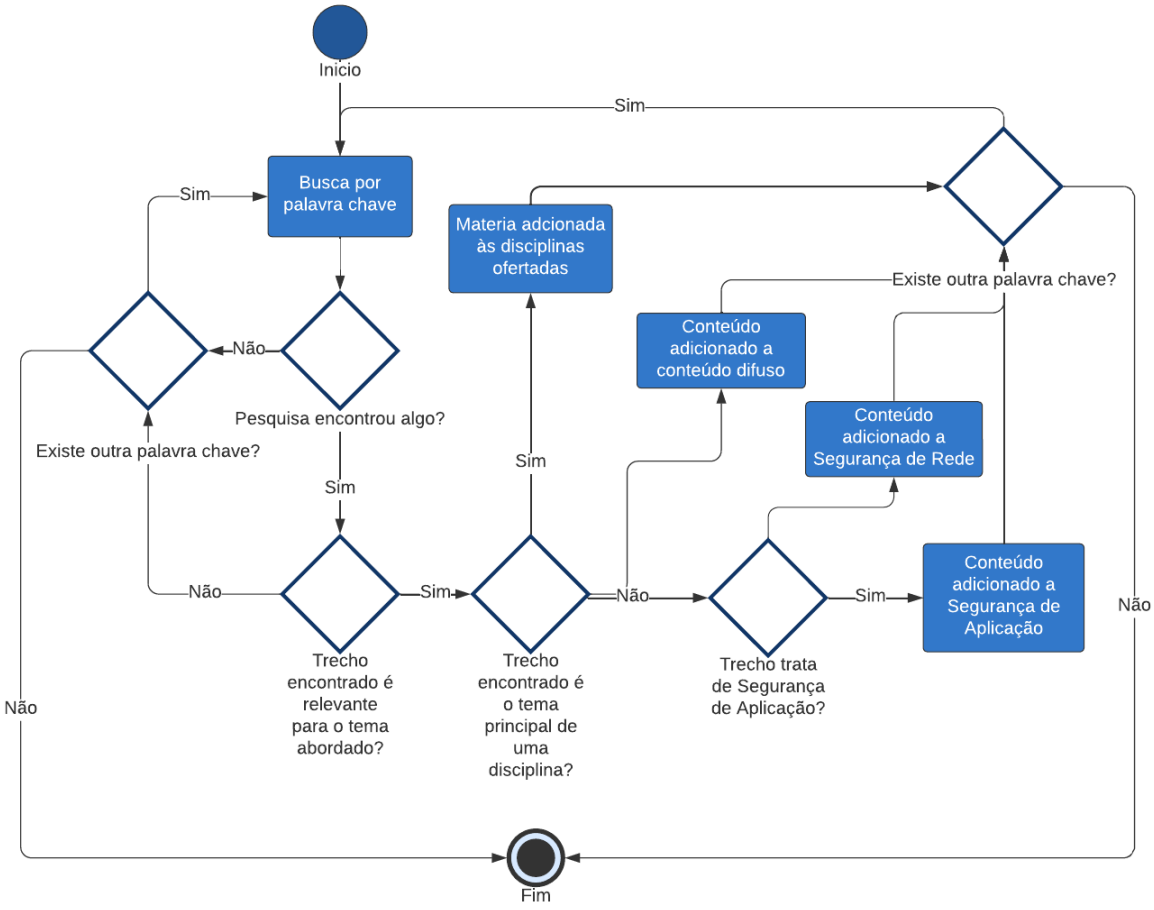
O processo começa com a análise do PPC das universidades brasileiras, documento que reúne todas as disciplinas e conteúdos. Em seguida, realizou-se uma pesquisa manual em cada documento, procurando por palavras-chave específicas. Este método permitiu identificar os tópicos relacionados à segurança de software abordados no programa pedagógico do curso de Engenharia de Software de cada universidade. As palavras-chave utilizadas foram:

- defensiva | defensive
- ameaça | threat
- confidencialidade | confidenciality
- segurança | security
- ataque | attack
- desenvolvimento seguro | secure coding
- vulnerabilidades | vulnerabilities
- privacidade | privacy
- proteção | protection
- crime

Para cada resultado desta busca são encontradas disciplinas ou assuntos que abordam a segurança, a partir disso, uma série de questionamentos são aplicados para categorizar estes resultados. O processo utilizado, bem como os questionamentos, podem ser

visualizados na Figura 9, onde cada busca é classificada de forma que identifica o seu impacto na formação dos profissionais. Os resultados desta organização são sumarizados na Tabela 1.

Figura 9 – Metodologia seguida para análise documental - Universidades brasileiras.




Os resultados da análise são organizados em relação às instituições de ensino (linhas). As colunas quantificam as seguintes informações:

- **Disciplinas ofertadas**: disciplinas voltadas de forma majoritária para a segurança, que podem ser utilizadas para compor a estrutura de formação do discente. Ex: Disciplinas Obrigatórias ou Disciplinas Optativas.
- **Conteúdo difuso**: disciplinas do currículo que apenas abrangem algum assunto ligado a segurança.
- **Segurança de rede/aplicação**: sendo encontrado alguma abordagem de segurança, aqui a abordagem é classificada conforme o seu foco, sendo este focado em segurança aplicada a camada de rede, ou segurança aplicada a camada de aplicação. Em alguns casos, uma disciplina pode ser categorizada como segurança de rede e segurança de aplicação, quando os dois temas possuem forte aplicação em uma disciplina.

## 4.2 Universidades Brasileiras

A Tabela 1 apresenta o resultado da análise das universidades brasileiras. Para cada universidade são apresentados dados sobre as suas disciplinas oferecidas que são predominantemente relacionados à segurança, assim como conteúdos de segurança tratados de forma difusa em seu currículo. Além disso, também é apresentada a divisão entre disciplinas com foco em segurança de redes e segurança de aplicação.

Nesta seção, foram coletados dados de instituições de ensino brasileiras que fornecem o curso de Engenharia de Software. Os critérios para a análise das universidades que oferecem o curso incluem:

- Disponibilidade da estrutura do curso por meio do PPC
- Oferta do curso no formato presencial
- O curso deve ser de nível bacharelado

Os dados apresentados na Tabela 1 demonstram carência na disponibilização de disciplinas voltadas para segurança de software na formação de Engenheiros de Software. Das universidades analisadas, quatro não oferecem opções de disciplina que desenvolvam os conhecimentos voltados à segurança. Os resultados podem ser verificados de forma integra no Apêndice A.

Tabela 1 – Análise dos cursos de Engenharia de Software em universidades brasileiras

| Universidades | Disciplinas ofertadas | Conteúdo difuso | Segurança de Rede | Segurança de Aplicação |
|---|---|---|---|---|
| UFMS | 5 | 9 | 5 | 10 |
| UNIPAMPA | 2 | 1 | 2 | 1 |
| UFC | 2 | 0 | 1 | 2 |
| UEPA | 1 | 3 | 3 | 1 |
| UnB | 1 | 3 | 2 | 1 |
| UFTPR | 1 | 1 | 0 | 3 |
| UFERSA | 1 | 0 | 1 | 1 |
| UFAM | 0 | 8 | 1 | 9 |
| UDESC | 0 | 5 | 1 | 4 |
| UFG | 0 | 4 | 1 | 3 |
| UFRN | 0 | 1 | 0 | 1 |

Na análise do conteúdo aplicado de forma difusa pelo curso, é possível observar a disparidade na quantidade de temas abordados entre segurança de aplicações e segurança de rede. Esse desequilibro pode indicar uma abordagem desbalanceada, com carência no desenvolvimento de habilidades nos assuntos menos recorrentes (CALIL, 2023).

É importante destacar que, para a classificação do foco da disciplina, serão utilizados os assuntos abordados de forma difusa em conjunto com as disciplinas com foco em segurança. É importante mencionar que, em alguns casos, as disciplinas abordam simultaneamente temas de segurança de rede e segurança de aplicação, o que resulta em valores maiores que o numero de disciplinas ofertadas.

## 4.3 Universidades referência

Para identificar as universidades cujos currículos são considerados referência para a formação de engenheiros de software, foi conduzida uma análise das instituições reconhecidas como as melhores em sua área, de acordo com fontes especializadas, incluindo edurank (2021), Education (2023) e USNEWS (2023). Nessa análise, foram escolhidos e comparados os 25 cursos listados no topo de cada ranking, e foram selecionadas as instituições que se encontravam consistentemente entre os melhores em todos os rankings consultados.

Levando em consideração a grande variedade de programas e contexto de mercado em um âmbito mundial, a comparação entre estas instituições com as universidades brasileiras se mostra extremamente difícil. Entre as dificuldades encontradas, é possível destacar:

- **Estrutura Curricular**: A estrutura dos cursos de bacharelado no Brasil tende a ser mais rígida e padronizada, com menos espaço para a escolha de disciplinas pelos alunos. Em contraste, no exterior, especialmente em países como França, Alemanha e Itália, os cursos oferecem maior flexibilidade, permitindo aos estudantes maior liberdade na seleção de suas matérias.

- **Duração do Curso**: Enquanto no Brasil a maioria dos cursos de bacharelado tem uma duração fixa de quatro ou cinco anos, em muitos países estrangeiros, como França, Alemanha e Itália, a duração padrão dos cursos é de três anos.

Tendo em vista as dificuldades existentes relacionadas à comparação dos cursos, foi decidido que a comparação será realizada com universidades dos *Estados Unidos da América* (EUA). Considerando que universidades dos EUA possuem as vantagens listadas a seguir:

- Melhor compreendimento dos dados devido ao idioma nativo.

- Maior facilidade no acesso a informações sobre o mercado de trabalho e estrutura dos cursos.

- Se trata do país com a maior incidência de universidades listadas no topo dos rankings das instituições utilizadas na pesquisa, com 14 no edurank (2021), 16 na Education (2023) e 10 no USNEWS (2023).

Considerando que a estrutura de formação utilizada será a dos EUA, é preciso esclarecer a diferença na formação de profissionais, uma vez que o tema levantado trata de cursos de Engenharia de Software.

**Formação do profissional**: segundo o *Bureau of Labor Statistics* (BLS) (STATISTICS, 2023), um profissional, com a intenção de atuar na área de engenharia de software, pode buscar o bacharelado em Ciência da Computação, Sistemas da Informação, Ciência de Dados, Redes de Comunicação, Engenharia ou Matemática, onde 61% dos profissionais escolhem o curso de Ciência da computação (STATISTICS; PROJECTIONS, 2021).

É importante ressaltar que algumas universidades elencadas no topo dos rankings não se encontram listadas na Tabela 2, isto ocorre devido a um dos motivos:

- **Indisponibilidade da lista de disciplinas**: o curso não oferece a lista de disciplinas do curso de forma aberta.
- **Disciplinas sem ementa**: a lista de disciplinas não possui os tópicos abordados na disciplina.

Considerando os pontos levantados, a Tabela 2 apresenta os resultados obtidos na pesquisa realizada para cursos de Ciência da Computação. Os resultados podem ser verificados de forma integra no Apêndice B.

É possível notar que existe uma discrepância notável entre a abordagem utilizada para as universidades analisadas, com algumas instituições oferecendo um extenso repertório, chegando a ter até 8 disciplinas que abordam do tema de segurança, enquanto outras não possuem essa escolha disponível para seus discentes. Quando o conteúdo é levado em consideração, é possível notar que a maioria das universidades aborda os temas de segurança de aplicação e segurança de rede em proporções similares, demonstrando a preocupação da preparação dos profissionais para lidar com problemas de diferentes camadas (CALIL, 2023).

## 4.4 Análise das universidades

Esta seção é dedicada a utilizar os dados coletados nas Seções 4.2 e 4.3, buscando comparar a abordagem utilizada em universidades brasileiras com instituições internacionais. É importante recordar que o levantamento de dados nas universidades brasileiras foi feito avaliando cursos de Engenharia de Software, enquanto que para as instituições

Tabela 2 – Análise de universidades estrangeiras para cursos de Ciência da Computação

| Universidades | Disciplinas ofertadas | Conteúdo difuso | Segurança de Rede | Segurança de Aplicação |
|---|---|---|---|---|
| Stanford | 8 | 11 | 8 | 10 |
| JHU | 8 | 4 | 6 | 7 |
| UChicago | 6 | 1 | 3 | 3 |
| Berkley | 4 | 8 | 3 | 7 |
| Carnegie Mellon | 4 | 1 | 1 | 4 |
| Harvard | 4 | 0 | 1 | 1 |
| Princeton | 2 | 3 | 3 | 4 |
| UW | 2 | 3 | 3 | 2 |
| MIT | 1 | 1 | 1 | 1 |
| Caltech | 0 | 1 | 0 | 1 |

estrangeiras foram analisados os cursos de Ciência da Computação. Isto se dá devido à diferença na natureza da estruturação da formação no exterior, onde os discentes têm maior liberdade para se especializar na área desejada, desde que esta se encontre acessível ao seu currículo.

Como demonstrado na Tabela 3, existem desigualdades significativas nas abordagens estudadas. Enquanto 10 universidades estrangeiras oferecem 39 disciplinas dedicadas à segurança, as 11 universidades brasileiras oferecem apenas 13 disciplinas. Além disso, conforme discutido nas seções anteriores, apenas uma das instituições estrangeiras não oferece opção para os discentes que procuram especialização na área de segurança, enquanto quatro das instituições brasileiras não disponibilizam disciplinas específicas na área. Por fim, é possível observar que os temas de segurança de rede e de aplicação possuem uma distribuição mais uniforme dentro da metodologia utilizada do exterior, quando comparado à nacional.

Tabela 3 – Comparação entre disciplinas de segurança em universidades brasileiras e estrangeiras

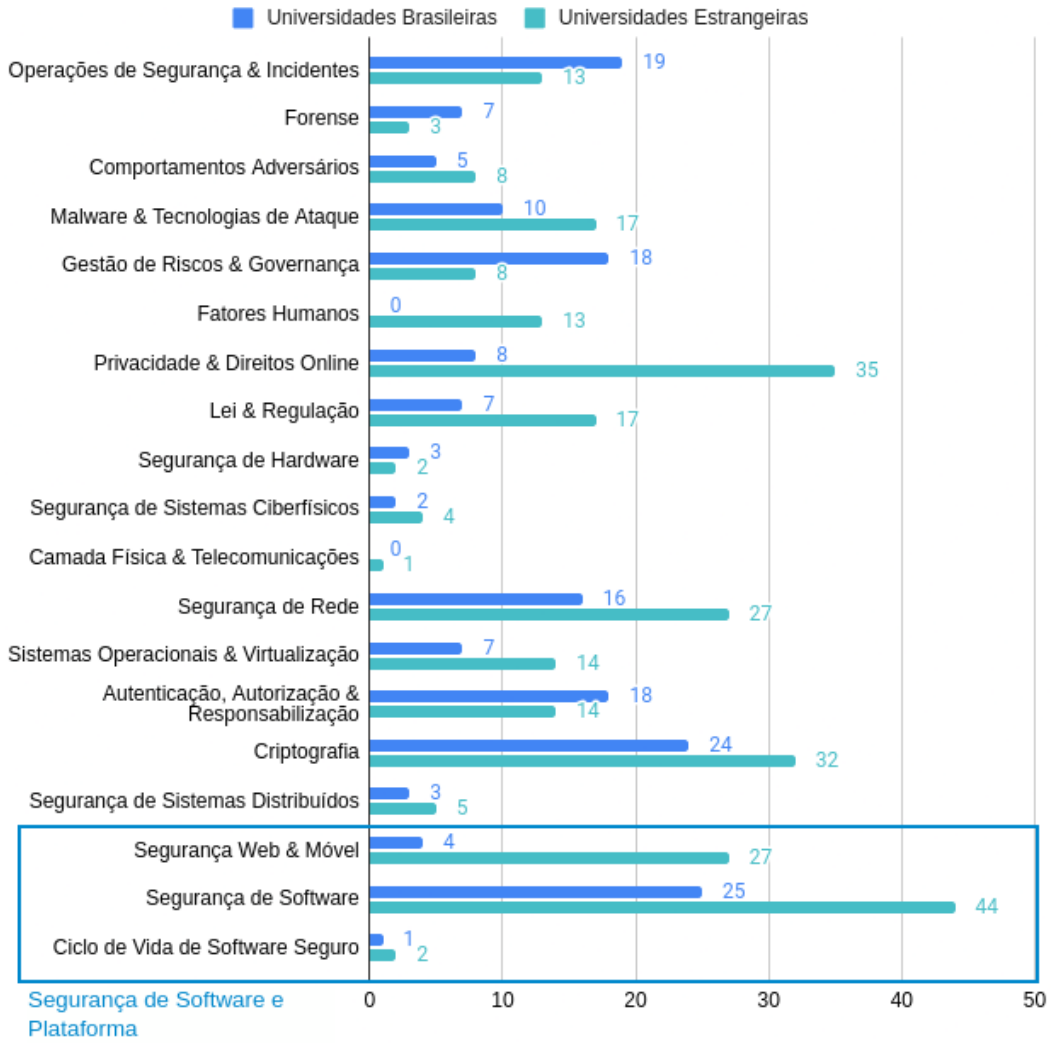
| **Critério** | **Brasileiras** | **Estrangeiras** |
|---|---|---|
| Número de Universidades | 11 | 10 |
| Total de Disciplinas Oferecidas | 13 | 39 |
| Universidades sem Disciplinas Específicas | 4 | 1 |
| Proporção entre temas de Segurança de Rede e Segurança de Aplicação | Menos Uniforme | Mais Uniforme |

Buscando compreender melhor as divergências no método de ensino entre as universidades brasileiras e estrangeiras, foi realizada a classificação de cada tópico encontrado que é voltado à segurança, seja ele dentro de uma disciplina ou conteúdo difuso. A classificação foi feita utilizando as 19 áreas de conhecimento especificadas no CyBOK (RASHID et al., 2019), onde cada assunto pode ser classificado em uma ou mais áreas, a depender

da ambiguidade ou abrangência dos termos utilizados. Com isso, é importante reconhecer que as informações levantadas nas ementas das disciplinas é limitada, o que impede uma classificação fidedigna do conteúdo abordado. Assim, as classificações são meramente aproximações do contexto real, visando prover apenas uma visão das tendências na linha de ensino dos cursos.

É importante ressaltar que a classificação utilizada para este gráfico é diferente da abordagem mais generalista utilizada nas Tabelas 1 e 2, onde ao invés de classificar a disciplina ou tópico como um todo, classificou-se cada tema proposto dentro das disciplinas.

Figura 10 – Comparação na abordagem de áreas de conhecimento da segurança cibernética entre instituições brasileiras e estrangeiras




Fonte: (CALIL, 2023)

A partir desta classificação, demonstrada na Figura 10, é possível visualizar a distribuição e o enfoque dado a determinados temas dentro da formação dos profissionais. A partir das informações recolhidas na classificação, pode-se destacar alguns pontos:

- **Áreas com maior foco**: É possível identificar uma priorização de determinadas assuntos em ambas as esferas de cobertura. Temas como Segurança de Software, Criptografia e Segurança de Redes são temas recorrentes com ampla abrangência em ambos os casos.
- **Diferenças na cobertura de Áreas de Conhecimento**: Existe uma disparidade notável entre as universidades brasileiras e estrangeiras em alguns temas. Nas instituições do exterior identifica-se um foco maior para assuntos que tratam da leis e regulamentação, fatores humanos, comportamentais, culturais e segurança web/mobile. Já as universidades brasileiras demonstram foco para temas relacionados a práticas de mitigação de risco.
- **Enfoque em segurança de Software**: Ambos conjuntos de universidades demonstram priorização aos temas de segurança de software, com as estrangeiras oferecendo uma gama mais abrangente de conteúdos.
- **Lacunas existentes**: Apesar de existirem diversas áreas pouco abordadas, em ambos os casos, destaca-se a inexistência de conteúdo disponível nos temas de Fatores Humanos e Telecomunicações & Camada Física nas universidades brasileiras.
- **Deficiência em áreas críticas**: É possível notar que existem áreas que possuem pouco foco em ambos os casos, mas considerando o objetivo deste trabalho, focaremos em assuntos cobertos na área de conhecimento de segurança de software e plataforma. Ambas as abordagens demonstram deficiências em sua abordagem do desenvolvimento de software seguro, um componente indispensável para a criação de produtos de software com segurança integrada. Outro ponto observado é da abordagem limitada das universidades nacionais quanto ao tema de segurança web e mobile.

## 4.5 Identificação de lacunas em comparação com diretrizes recomendadas

Após realizar a comparação entre as universidades, o próximo passo é de considerar as diretrizes propostas por estudos e instituições renomadas. Este levantamento procura verificar se as práticas pedagógicas utilizadas atualmente são adequadas, quando levado em conta recomendações especializadas. Além disso, a análise também procura identificar possíveis melhorias, para elencar os pontos necessários para que o formato de ensino atual esteja alinhado com as práticas mais recentes e eficazes para o ensino de segurança de software.

Com base nas metodologias propostas pelo artigo de Yang, Lodgher e Lee (2018) e as orientações da IEE e ACM (2015), a Tabela 4 analisa a abordagem de universidades

brasileiras. Para esta análise, utilizou-se da listagem de assuntos voltados para a segurança nas universidades brasileiras. Essa análise busca verificar se a instituição possui alguma abordagem ao tema, mesmo que tratado em disciplinas que não possuem foco primário em segurança.

Tabela 4 – Checagem dos temas de segurança nas universidades brasileiras

| **Topicos** | **UnB** | **UFC** | **UFG** | **UEPA** | **UFMS** | **UFAM** | **UFRN** | **UDESC** | **UFTPR** | **UFERSA** | **UNIPAMPA** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Tema 1 | | | | | X | X | | | | | |
| Tema 2 | | | X | | X | X | | | | X | |
| Tema 3 | X | X | X | | X | | X | | | X | |
| Tema 4 | | | | | | X | | | | | |
| Tema 5 | | X | X | | | | | | X | | X |
| Tema 6 | | | | X | X | | | | | | X |
| Tema 7 | X | | | X | X | | | X | X | X | X |
| Tema 8 | | X | | X | X | | | | X | X | X |
| Tema 9 | X | X | | X | X | | | X | | X | X |
| Tema 10 | X | X | | X | X | | | X | X | X | X |
| Tema 11 | | | | | X | | | X | | | |
| Tema 12 | X | X | | X | X | | | | | X | |
| Tema 13 | X | X | | X | X | | | | | X | X |
| Tema 14 | X | X | | X | X | | | | | | X |

Os temas listados na Tabela 4 se referem aos temas propostos pelo artigo de Yang, Lodgher e Lee (2018) e as orientações da IEE e ACM (2015), conforme descrito abaixo.

1. Análise e especificação de requisitos de segurança;
2. Princípios e padrões do design de segurança;
3. Técnicas e boas práticas para desenvolvimento de software seguro;
4. Verificação e validação de segurança;
5. Aplicar manejamento de risco no plano do projeto de software;
6. Uso de ferramentas de engenharia reversa para realizar analises estáticas e dinâmicas em software ou malware de fontes desconhecidas;
7. Conceitos de segurança da informação(confidencialidade, integridade a disponibilidade);
8. Natureza das ameças;
9. Encriptação, assinaturas digitais, autenticação de mensagens e funções hash;
10. Protocolos de criptografia mais comuns;
11. Engenharia social;
12. Segurança de redes: Ameças e ataques;
13. Uso de criptografia para segurança de redes;
14. Proteção e mecanismos de defesa para segurança de redes.

A partir da visualização da abordagem gerada na Tabela 4, é possível observar os pontos abordados e omitidos no método de ensino das instituições brasileiras. Pontos que incluem:

- **Abordagem limitada para áreas importantes**: Os temas listados de 1 a 4 são mencionados em ambos os estudos. É possível identificar carência na abordagem destes, com apenas uma universidade tratando a **Verificação e validação de segurança** e apenas duas universidades tratando da **Análise e especificação de requisitos de segurança**.
- **Grande disparidade nas abordagens**: Enquanto existem universidades como a UFMS que aborda grande parte dos pontos listados nas diretrizes, existem universidades que incluem poucas destas práticas em sua estrutura de ensino, a UFRN, UFG e UFAM.

- **Foco em segurança de Redes**: Os assuntos voltados para a segurança de redes possuem uma incidência maior que os voltados para outras áreas. É possível demonstrar isso com o assunto de **protocolos de criptografia mais comuns** (Tema 10), que foi o mais abordado, com oito universidades possuindo algum nível de abordagem dentro do currículo do curso. Também podemos ver que os temas de **Encriptação, assinaturas digitais, autenticação de mensagens e funções hash** (Tema 9) e **Uso de criptografia para segurança de redes** (Tema 13) foram abordados em mais da metade das instituições pesquisadas.

# 5 Questionário

O questionário foi utilizado com o intuito de coletar a percepção dos atuais e futuros profissionais sobre a importância da segurança de software no currículo do curso de Engenharia de Software. Esta seção introduz o questionário, destacando sua relevância e papel no contexto mais amplo da pesquisa sobre a eficácia dos currículos de Engenharia de Software no Brasil em incorporar aspectos essenciais de segurança de software.

## 5.1 Modelagem do instrumento

Uma vez que o propósito do questionário foi definido, os esforços foram voltados para a estruturação do instrumento. Levando em consideração seu objetivo, o documento foi dividido em cinco seções:

- **Termo de consentimento e Elegibilidade dos participantes**: Inicialmente, o questionário apresenta um termo de consentimento, esclarecendo os objetivos do estudo, os procedimentos a serem seguidos, a confidencialidade das informações e o caráter voluntário da participação. Seguido pela caracterização do participante como aluno do curso de Engenharia de Software na UnB.

- **Percepção sobre a Importância da Segurança de Software**: Esta seção inclui perguntas que avaliam a importância da segurança de software na formação do engenheiro, abrangendo desde a classificação da relevância de diferentes áreas do SWEBOK até a avaliação da importância da segurança em diferentes contextos, utilizando escalas Likert. Uma questão adicional avalia a satisfação dos alunos com a abordagem atual da segurança de software no currículo.

- **Segurança de Software no Currículo de Engenharia de Software na UnB**: Disponível somente para participantes que consideram insuficiente a abordagem atual. Esta seção foca em entender as preferências dos alunos quanto à implementação da segurança de software no currículo, incluindo a escolha da abordagem curricular e a seleção dos tópicos mais importantes.

- **Questões Demográficas**: Duas perguntas demográficas buscam classificar os participantes conforme o ano de ingresso no curso e a experiência profissional no mercado de trabalho.

- **Experiência de Mercado**: Esta seção é dedicada a avaliar a experiência prática dos alunos com a segurança de software no mercado de trabalho, incluindo questões

sobre a aplicação de práticas de segurança, a confiança dos participantes quando executam tarefas relacionadas à segurança e treinamentos recebidos fora da universidade.

O instrumento foi criado utilizando a plataforma Microsoft Forms (MICROSOFT, 2023a), uma vez que a plataforma é disponibilizada de forma gratuita para alunos da UnB. Além disso, a plataforma também disponibiliza recursos que foram importantes para a divulgação, como a restrição que apenas pessoas que possuem um e-mail da UnB podem responder o questionário.

Após a confecção do instrumento, um subgrupo de cinco indivíduos da população foi selecionado para a realização de um piloto. O questionário foi avaliado em questões de compreensão, clareza e fluxo das questões, assim como a duração do mesmo. A partir do feedback dos participantes, foi possível identificar falhas no fluxo das seções e também problemas com a clareza do enunciado de algumas perguntas, a partir disso foi gerada uma versão definitiva do instrumento, observável na Tabela 5.

Tabela 5 – Versão final do questionário

| ID | Pergunta | Tipo de Resposta | Objetivo |
|---|---|---|---|
| Q1 | Termo de consentimento. | Binária | Consentimento para participação na pesquisa |
| Q2 | Você é aluno do curso de Engenharia de Software? | Binária | Identificar participante como parte da população alvo |
| Q3 | Ordene as áreas de conhecimento conforme a sua visão de importância para a construção de um produto de software. | Ordenação | Qual prioridade é atribuída à segurança em relação a outras áreas da Engenharia de Software pelos participantes |
| Q4 | Qual a importância da segurança de software como uma competência de um profissional de Engenharia de Software? | Escala Likert | Avalia a importância que os participantes dão para a capacidade de lidar com problemas de segurança |
| Q5 | Quão importante você considera que o tema de segurança de software seja tratado durante a formação de um Engenheiro de Software? | Escala Likert | Qual a importância no Currículo de formação |
| Q6 | Em aplicações de software que você utiliza, o quanto você considera os aspectos de segurança de software? | Escala Likert | Coletar a visão dos participantes quando no papel de usuários de aplicações de software |
| Q7 | Em aplicações de software que você desenvolve, o quanto você considera os aspectos de segurança de software? | Escala Likert | Coletar a visão dos participantes quando no papel de desenvolvedores de aplicações de software |
| Q8 | Você considera que o currículo do curso de Engenharia de Software na UnB dá atenção adequada para o tema de segurança? | Múltipla Escolha | Verificar a satisfação com o currículo vigente |

**Tabela 5**

| ID | Pergunta | Tipo de Resposta | Objetivo |
|---|---|---|---|
| Q9 | Na sua opinião, qual seria a implementação mais adequada para abordar o tema de segurança no curso de Engenharia de Software? | Múltipla Escolha | Identificar a preferência na implementação de abordagens ao tema de segurança de software |
| Q10 | Marque os cinco tópicos voltados à segurança de software que você considera mais importantes de serem abordados no curso de Engenharia de Software. | Escolha Múltipla | Conhecer os tópicos mais interessantes na visão dos participantes |
| Q11 | Qual o ano que você ingressou no curso de Engenharia de Software? | Múltipla Escolha | Verificar o estágio do curso que os participantes estão |
| Q12 | Você já atua no mercado de trabalho como engenheiro de software? | Binária | Experiência de Mercado |
| Q13 | Quanto tempo de experiência profissional você possui na área de desenvolvimento de software? | Múltipla Escolha | Tempo de Experiência |
| Q14 | Você já aplicou alguma prática de segurança de software no seu trabalho? | Binária | Determinar a frequência de demandas por funcionalidades de segurança de software aos profissionais. |
| Q15 | Se você for atribuído a alguma tarefa relacionada à segurança da aplicação, você se considera seguro(a) para executá-la? | Binária | Confiança no conhecimento adquirido |
| Q16 | Você já recebeu algum tipo de treinamento (fora da universidade) para aplicar práticas de segurança durante o ciclo de desenvolvimento de software? | Binária | Identificar se o conhecimento adquirido em segurança de software é proveniente de cursos feitos fora da universidade |
| Q17 | Você tem alguma dúvida ou sugestão? | Aberta | Feedback |

## 5.2 Divulgação

Com o instrumento validado, iniciou-se a fase de divulgação do questionário. A disseminação ocorreu por meio de redes de comunicação, utilizadas por discentes e ex-alunos do curso de Engenharia de Software da UnB, mais especificamente um grupo de Telegram com 1478 membros. É importante ressaltar que, dada a natureza aberta deste grupo, não é possível assegurar que todos os membros sejam estudantes ou ex-estudantes da UnB. A partir desta fase de divulgação, obteve-se aproximadamente 30 respostas, um número consideravelmente baixo.

Em seguida, adotou-se uma nova estratégia de divulgação, na qual o responsável pela pesquisa reforçou o convite aos participantes da primeira fase, juntamente com um novo esforço de divulgação realizado pela orientadora do trabalho, que compartilhou o questionário com suas turmas na universidade. Esta última abordagem mostrou-se eficaz, resultando em 30 novas participações e totalizando 60 respostas únicas ao questionário.

## 5.3 Resultados

Após aproximadamente duas semanas de divulgação, foi notado que a taxa de participação diminuiu significativamente, com isso decidiu-se por encerrar a etapa de coleta de respostas. Durante a aplicação, nenhum participante relatou dificuldades ou incerteza sobre a pesquisa.

Após o término da coleta, iniciou-se a análise dos dados, excluindo respostas inconsistentes, como as de alunos da UnB que não cursam Engenharia de Software, conforme identificado na Questão 2.

### 5.3.1 Opinião dos alunos sobre a abordagem do curso na UnB

Conforme exposto na Figura 11 93% dos respondentes, declarou que iniciou o curso antes de 2022. Dado que o questionário foi distribuído no final de 2023 e o curso tem duração total estimada de dez semestres, pressupõe-se que a maioria dos participantes já completaram mais da metade do programa. Assim, presume-se que esses participantes possuem conhecimento suficiente, para avaliar criticamente a abordagem da segurança de software no curso de engenharia de software na UnB.

Dado que os participantes possuem conhecimento adequado sobre o programa oferecido, a Figura 12 demonstra que existe uma clara insatisfação dos envolvidos quanto à abordagem utilizada no curso atualmente para o tema de segurança, com apenas um indivíduo considerando a metodologia atual satisfatória, contrapondo os 98,3% que consideram a dinâmica atual inadequada.

Figura 11 – Ano de ingresso dos participantes no curso de Engenharia de Software

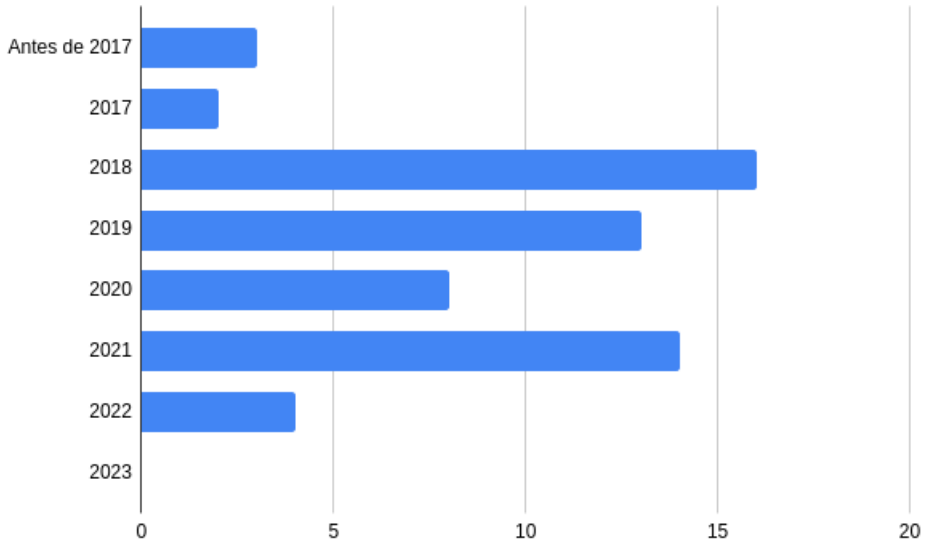
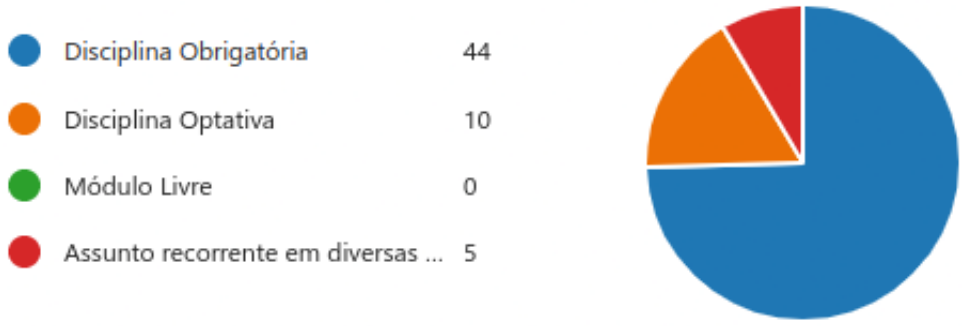



Figura 12 – Você considera que o currículo do curso de Engenharia de Software na UnB dá atenção adequada para o tema de segurança?


Considerando apenas os participantes que indicaram a insatisfação com a atual abordagem, 75% demonstraram o interesse em uma disciplina obrigatória, 17% em uma disciplina optativa e os remanescestes 13% preferem que o tema seja abordado em diversas disciplinas ao decorrer do curso, proporção que pode ser visualizada na Figura 13.

Figura 13 – Na sua opinião, qual seria a implementação mais adequada para a abordar o tema de segurança no curso de Engenharia de Software?

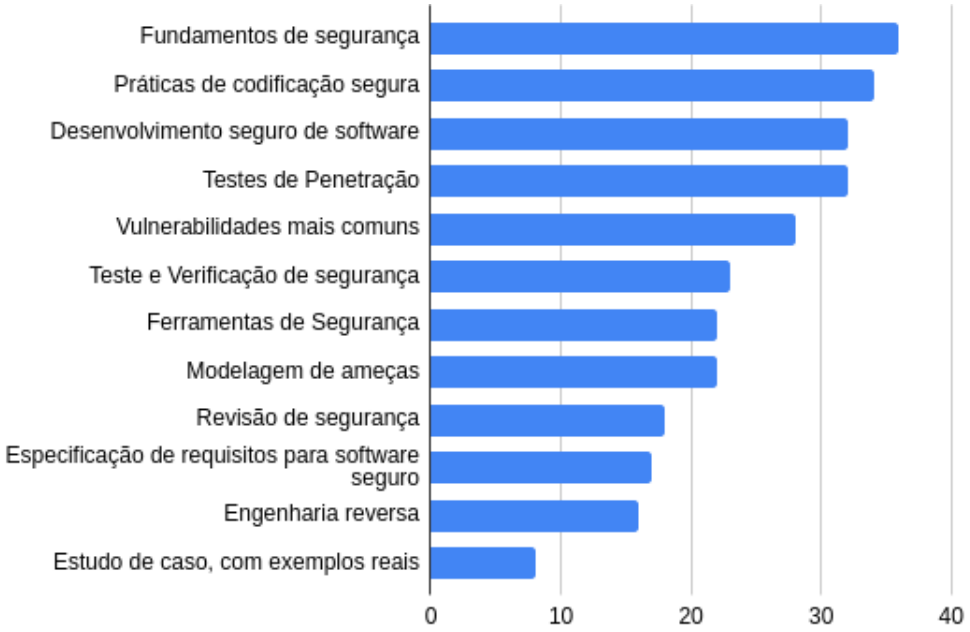



Em seguida, os participantes escolheram até cinco temas, em uma lista com dez possibilidades, para compor a abordagem da segurança de software no curso. A Figura

14 apresenta quatro temas em destaque, **Fundamentos de Segurança**, **Práticas de codificação segura**, **Desenvolvimento de software seguro** e **Testes de penetração**, cada um destes com mais de 30 votos.

O resultado demonstra grande interesse dos participantes em práticas voltadas para a área de Segurança de Software e de Plataforma, que é composto pelos assuntos voltados à Segurança de Software, Desenvolvimento Seguro de Software e Segurança Web & Mobile. Além disso, as áreas mais votadas também estão alinhadas às diretrizes propostas pelos estudos mencionados na Seção 4.5.

Figura 14 – Temas mais importantes na visão dos respondentes




### 5.3.2 A importância da segurança de software

O estudo investigou a priorização da segurança pelos participantes em comparação com outras disciplinas da Engenharia de Software. A Figura 15 ilustra como os estudantes envolvidos classificam a segurança em relação às áreas de conhecimento definidas pelo SWEBOK (ABRAN, 2005). Visando melhorar a interpretação dos participantes e também incluir a segurança como uma opção, algumas alterações foram realizadas nas áreas apresentadas no SWEBOK, sendo elas representadas na Tabela 6.

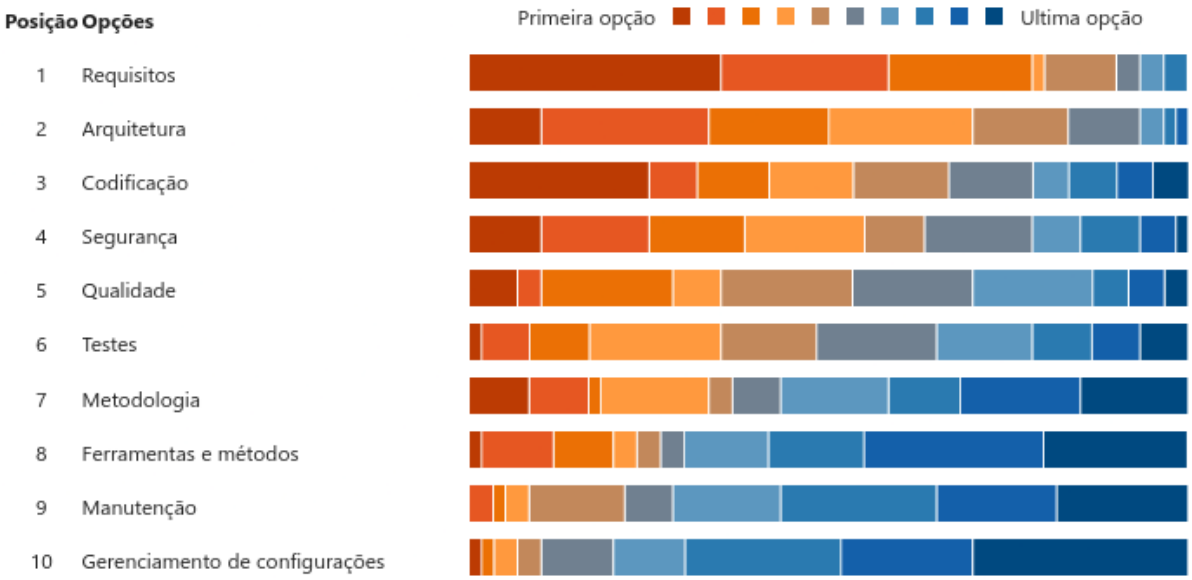
É possível observar que, mesmo com áreas como Requisitos, Arquitetura de Software e Codificação sejam consideradas mais relevantes, a Segurança ainda foi classificada como mais importante que assuntos como Qualidade, Testes e Metodologia pelos respondentes.

Em conjunto com a priorização, os envolvidos avaliaram o valor que eles atribuem à segurança de software em diferentes cenários, como demonstrado na Figura 16 em conjunto

Tabela 6 – Correspondência de Termos entre SWEBOK e Questionário.

| **Termo utilizado pelo SWEBOK** | **Termo no Questionário** |
|---|---|
| Software Requirements | Requisitos |
| Software Design | Arquitetura |
| Software Construction | Codificação |
| Software Testing | Testes |
| Software Maintenance | Manutenção |
| Software Configuration Management | Gerenciamento de Configurações |
| Software Engineering Management | Segurança |
| Software Engineering Process | Metodologia |
| Software Engineering Tools and Methods | Ferramentas e Métodos |
| Software Quality | Qualidade |

Figura 15 – Ranqueamento de áreas de conhecimento da Engenharia de Software




com a Tabela 7. Dentre as opções disponíveis, considera-se que uma pontuação de sete a dez indica uma alta importância para os participantes.

A segurança foi considerada uma competência muito importante por 50 (83%) dos respondentes para a atuação de profissionais da Engenharia de Software. Na formação de engenheiros, 44 (73%) deram alta importância para que a segurança seja tratada durante a formação. Quanto aos produtos de software utilizados por eles, a segurança foi valorizada por 42 (70%), e, por fim, apenas 22 (36%) dos participantes informou considerar aspectos de segurança enquanto está desenvolvendo aplicações de software. Os dados mostram que, embora haja uma alta valorização da segurança de software em ambientes acadêmicos e de uso no dia-a-dia, a área recebe menos atenção dos participantes durante o processo de desenvolvimento de aplicações.

Figura 16 – Questões sobre a importância do tema de segurança de Software.

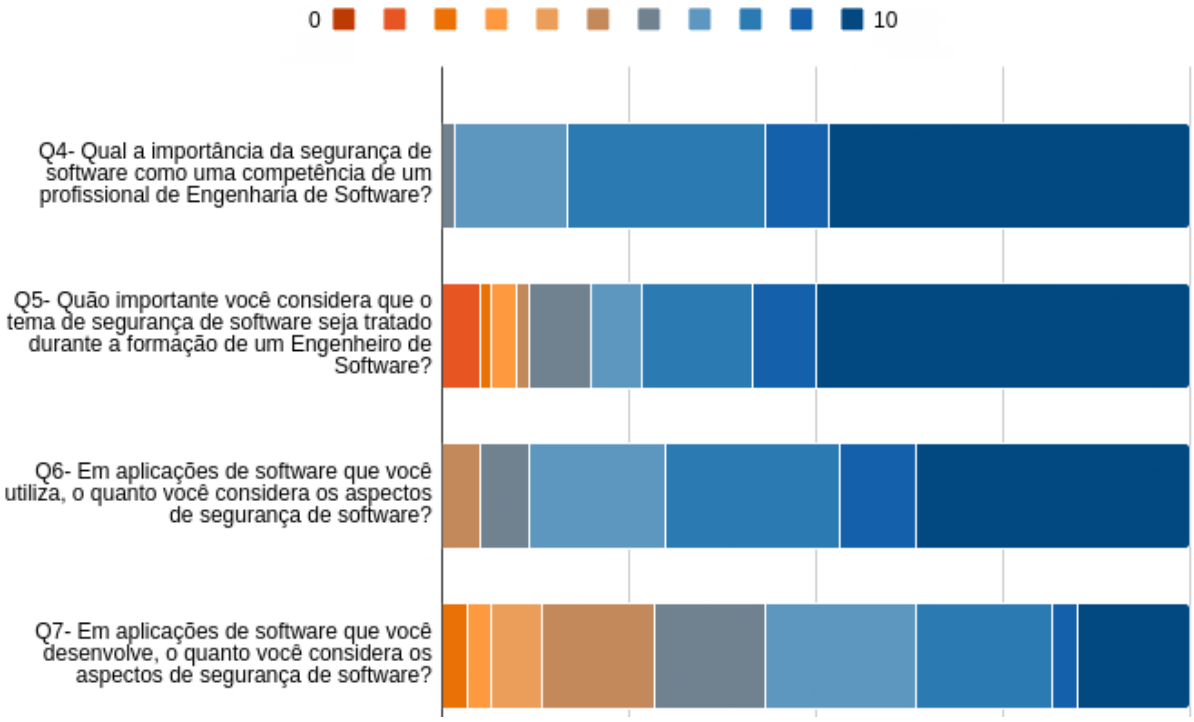



Tabela 7 – Avaliação da Importância da Segurança de Software na Engenharia de Software

| **Questão** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Q4- Qual a importância da segurança de software como uma competência de um profissional de Engenharia de Software? | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 9 | 16 | 5 | 29 |
| Q5- Quão importante você considera que o tema de segurança de software seja tratado durante a formação de um Engenheiro de Software? | 0 | 3 | 1 | 2 | 0 | 1 | 5 | 4 | 9 | 5 | 30 |
| Q6- Em aplicações de software que você utiliza, o quanto você considera os aspectos de segurança de software? | 0 | 0 | 0 | 0 | 0 | 3 | 4 | 11 | 14 | 6 | 22 |
| Q7- Em aplicações de software que você desenvolve, o quanto você considera os aspectos de segurança de software? | 0 | 0 | 2 | 2 | 4 | 9 | 9 | 12 | 11 | 2 | 9 |

### 5.3.3 Experiência com segurança no mercado de trabalho

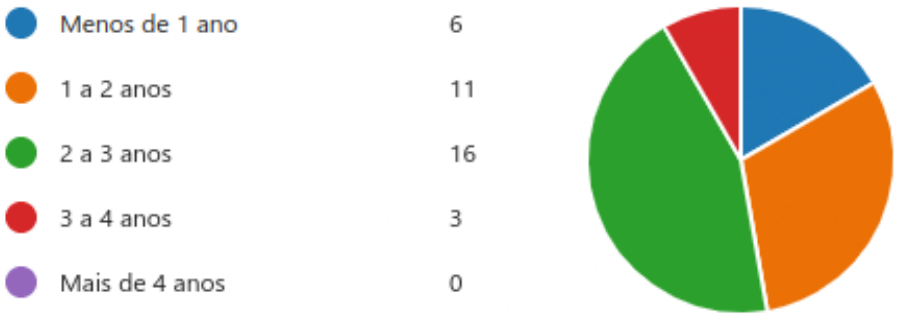
Levando em conta as experiências dos participantes, como demonstrado na Tabela 8, o levantamento apontou que 60% dos participantes já tiveram contato com o mercado de trabalho. Destes, a maioria (83%) não se sentem confortáveis para implementar práticas de segurança, mesmo que, como demonstrado na Figura 17, 83% dos respondentes declararam possuir mais de um ano de experiência profissional e quase metade deles (47%) declararam que já precisaram implementar alguma prática de segurança no trabalho.

Tabela 8 – Experiência e Confiança em Práticas de Segurança de Software

| **Perguntas** | **Sim** | **Não** |
|---|---|---|
| Q12- Você já atua no mercado de trabalho como engenheiro de software (estágio, CLT, PJ, MEI) | 36 | 24 |
| Q14- Você já aplicou alguma prática de segurança de software no seu trabalho? Por exemplo: modelagem de ameaças, práticas de codificação segura, testes de segurança, etc. | 17 | 19 |
| Q15- Se você for atribuído à alguma tarefa relacionada à segurança da aplicação, você se considera seguro(a) para executá-la? | 6 | 30 |
| Q16- Você já recebeu algum tipo de treinamento (fora da universidade) para aplicar práticas de segurança durante o ciclo de desenvolvimento de software? | 12 | 24 |

Considerando que 33% deles declararam já terem recebido algum tipo de treinamento além do que viram na universidade, os resultados demonstrados na Tabela 8 indicam que os estudantes/profissionais ainda não possuem confiança na implementação de práticas de segurança de software. O resultado sugere uma deficiência prática no enfrentamento de desafios específicos na área, o que pode impactar negativamente a autoconfiança dos profissionais em suas habilidades relacionadas à segurança de software.

Figura 17 – Tempo de experiência dos participantes



# 6 Recomendações

Este capítulo propõe recomendações para aprimorar o ensino de segurança de software nas universidades brasileiras, com base em diretrizes internacionais, práticas de universidades estrangeiras e preferências dos alunos brasileiros representados na pesquisa. O objetivo é de recomendar temas relevantes para um currículo equilibrado que prepare os alunos para os desafios reais da área de Engenharia de Software.

O levantamento de dados foi realizado buscando identificar o que é, ou deveria ser abordado no ensino da segurança de software. A partir dos resultados, foi possível construir uma lista de tópicos que receberam destaque em cada campo investigado. As próximas seções irão expor os temas encontrados e ao fim deste capítulo, será possível entender quais assuntos são mais relevantes, levando em consideração a recorrência dos assuntos nas áreas exploradas.

## 6.1 Diretrizes levantadas de artigos e instituições renomadas

A partir da análise feita na Seção 4.5, os artigos estudados sugerem que diversos temas devem ser abordados durante o ensino da segurança de software em um curso de Engenharia de Software. Entre os tópicos indicados, é possível apontar quatro temas que estão presente em ambos os artigos, assim como assuntos que também se encontram nas outras fontes de dados. Os conteúdos mencionados são:

- Análise e especificação de requisitos de segurança;
- Princípios e padrões do design de segurança;
- Desenvolvimento de software seguro;
- Verificação e validação de segurança;
- Conceitos de segurança da informação(confidencialidade, integridade a disponibilidade);
- Natureza das ameças;

## 6.2 Temas em comum na abordagem de universidades do exterior

Após a etapa de análise e caracterização dos tópicos de segurança, abordados em universidades estrangeiras, os assuntos encontrados foram agrupados com base em

assuntos focados em segurança de software, com a intenção de encontrar os assuntos mais recorrentes. A lista abaixo demonstra o resultado do agrupamento:

- Práticas de programação segura;
- Desenvolvimento seguro de software;
- Segurança em compiladores;
- Estudo de vulnerabilidades e defesas;
- Fundamentos de segurança;
- Ferramentas e técnicas de análise de software;
- Linguagens de programação segura.

## 6.3 Preferência dos alunos

Lista de assuntos recolhidos como um dos resultados da pesquisa. Dentro da Questão 5 os participantes escolheram temas que consideram importantes na abordagem de segurança dentro do curso de Engenharia de Software da UnB, dentre as dez opções, as seis que receberam mais votos (considerando o total de 59 respondentes) foram:

- Fundamentos de segurança **(36 votos)**;
- Práticas de codificação segura **(34 votos)**;
- Testes de penetração **(32 votos)**;
- Desenvolvimento Seguro de Software **(32 votos)**;
- Vulnerabilidades mais comuns **(28 votos)**;
- Teste e verificação de segurança **(23 votos)**.

## 6.4 Comparação

Com os tópicos de cada área pesquisada elencados, é possível encontrar semelhanças entre eles. Para que a representação possa ser melhor demonstrada, alguns temas tiveram seus nomes generalizados para se encaixarem melhor na organização. A Tabela 9 apresenta a relação entre os nomes utilizados aqui e seus nomes de origem em suas áreas específicas.

Tabela 9 – Unificação dos termos utilizados nos diferentes contextos de pesquisa

| **Nome original** | **Nome apresentado** | **Área de origem** |
|---|---|---|
| Análise e especificação de requisitos de segurança | Requisitos de segurança | Artigos e diretrizes de instituições renomadas |
| Princípios e padrões do design de segurança | Design de segurança | Artigos e diretrizes de instituições renomadas |
| Técnicas e boas praticas para desenvolvimento de software seguro | Desenvolvimento Seguro de Software | Artigos e diretrizes de instituições renomadas |
| Conceitos de segurança da informação | Fundamentos de Segurança | Artigos e diretrizes de instituições renomadas |
| Natureza das ameças | Estudo de vulnerabilidades e defesas | Artigos e diretrizes de instituições renomadas |
| Vulnerabilidades mais comuns | Estudo de vulnerabilidades e defesa | Questionário |
| Teste e verificação de segurança | Verificação e Validação de segurança | Questionário |

Figura 18 – Distribuição dos assuntos de segurança de software

No diagrama apresentado pela Figura 18, observamos uma variedade de tópicos relacionados à segurança de software distribuídos nas diferentes áreas pesquisadas. É notável que existem temas comuns a múltiplas áreas, indicando sua relevância no campo da segurança de software. Esses tópicos recorrentes são considerados importantes, pois estão presentes tanto nas estratégias de segurança adotadas atualmente, quanto nas orientações profissionais e no interesse acadêmico dos estudantes. Assim, a frequência com que estes temas aparecem no diagrama sinaliza a sua significância e a necessidade de atenção contínua dentro do domínio da segurança de software.

Com base nos dados apresentados neste capítulo, em conjunto com a preferência demonstrada pelos alunos, como demonstrado na Figura 13 é recomendado que seja criada a oferta de uma disciplina obrigatória com foco em segurança de software no currículo do curso de Engenharia de Software na UnB. Levando em conta o conteúdo programático da disciplina, aconselha-se a utilização dos temas que se demonstraram importantes no levantamento de dados realizado. Os temas são:

- Estudo de vulnerabilidades e defesas;
- Fundamentos de segurança;
- Desenvolvimento seguro de software;
- Práticas de programação segura;
- Verificação e Validação de segurança.

É importante ressaltar que as sugestões levantadas representam a coletânea de cenários levando às possibilidades mais recomendadas. Assim, a possível implementação das recomendações aqui citadas, deve ter sua viabilidade bem estudada antes da sua implementação em uma estrutura curricular.

# 7 Conclusão e Trabalhos futuros

Este trabalho apresenta os resultados de uma análise da abordagem no ensino de segurança de software nos cursos de Engenharia de Software. A análise foi realizada entre as estruturas curriculares de cursos oferecidos por universidades brasileiras, comparando a diretrizes sugeridas por artigos e instituições renomadas, estruturas curriculares de universidades estrangeiras e, por fim, às expectativas dos alunos do curso de Engenharia de Software da UnB.

O principal propósito da pesquisa foi de identificar possíveis melhorias ou falhas nas abordagens utilizadas atualmente nas universidades públicas brasileiras. Possíveis lacunas no ensino que podem impedir que os seus egressos exerçam de forma adequada as tarefas que são requisitadas deles no mercado de trabalho, que vem demandando cada vez mais que os profissionais tenham conhecimentos em segurança de software.

Para entender a abordagem utilizada atualmente, foram levantados dados das universidades brasileiras e estrangeiras, assim como o estudo de artigos e diretrizes que propõe a estruturação do ensino de segurança nas universidades. A partir dos dados recolhidos foi possível identificar pontos de importância, recorrentes em diversas áreas pesquisadas. Foi possível comparar as diretivas encontradas com as estruturas curriculares estudadas, o que demonstrou que a abordagem nacional possui falhas, mas apesar disso, algumas universidades possuem procedimentos que abordam grande parte das recomendações encontradas. É importante conseguir apontar estas possíveis lacunas nas estruturas dos cursos de Engenharia de Software, para que as universidades possam estudar abordagens que podem melhorar a capacitação dos seus egressos para o mercado futuro.

Em conjunto com o estudo dos modelos atuais, uma pesquisa foi realizada com os alunos de Engenharia de Software na UnB. Nesta pesquisa foi possível coletar informações sobre a satisfação dos participantes quanto à abordagem utilizada na sua universidade e qual a abordagem que eles gostariam de ter, qual a sua preocupação com a segurança dos aplicativos de software que desenvolvem e utilizam e também a cobrança que eles veem nos seus empregos. A partir destes resultados, foi possível conhecer as expectativas dos discentes e também entender o quanto isso é importante e requisitado deles no dia-a-dia enquanto profissionais.

A partir dos resultados encontrados, o trabalho traz como resultado a construção de recomendações, tanto para os tópicos que podem ser tratados, quanto para como estes assuntos devem ser incluídos na estrutura de um curso de Engenharia de Software.

Com a pesquisa da situação atual do ensino de segurança de software realizada neste trabalho, espera-se que trabalhos futuros possam estudar os cenários para a implan-

tação das recomendações encontradas. Considerando a grande diversidade de circunstâncias das instituições de ensino, é preciso que um plano de implantação seja exaustivamente estudado. Este plano deve considerar cuidadosamente as especificidades de cada universidade, buscando integrar a segurança de software ao currículo de forma equilibrada, evitando sobrecarga para alunos e professores, e assegurando que a instituição esteja adequadamente preparada para esta nova abordagem.

# Referências

ABRAN, A. (Ed.). *Guide to the software engineering body of knowledge: 2004 version ; SWEBOK*. Los Alamitos, Calif: IEEE Computer Soc, 2005. ISBN 978-0-7695-2330-9. Citado na página 61.

AKALA, A. *More big employers are talking about permanent work-from-home positions*. 2020. Disponível em: <https://www.cnbc.com/2020/05/01/major-companies-talking-about-permanent-work-from-home-positions.html>. Citado na página 19.

ANDRION, R. Mercado de cibersegurança cresce no brasil em meio à pandemia. 2021. Disponível em: <https://canaltech.com.br/seguranca/mercado-de-ciberseguranca-cresce-no-brasil-em-meio-a-pandemia-186631/>. Citado na página 20.

BBC. Coronavirus: Twitter allows staff to work from home 'forever'. *BBC News*, maio 2020. Disponível em: <https://www.bbc.com/news/technology-52628119>. Citado na página 19.

BLAIR, J. R. S. et al. Infusing Principles and Practices for Secure Computing Throughout an Undergraduate Computer Science Curriculum. In: *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2020. (ITiCSE '20), p. 82–88. ISBN 978-1-4503-6874-2. Disponível em: <https://doi.org/10.1145/3341525.3387426>. Citado na página 30.

CALIL, F. M. *Classificação de assuntos*. 2023. Disponível em: <https://docs.google.com/spreadsheets/d/1VR_dD5MBOT3_d5TpwWGQF0s5PtdEeBhJJGk67pNbKXM/edit?usp=sharing>. Citado 3 vezes nas páginas 45, 47 e 49.

CETIC.BR|NIC.BR. Privacy and data protection during the pandemic. 2022. Disponível em: <https://cetic.br/media/docs/publicacoes/6/20211217114412/iso-year-xiii-n-4-privacy.pdf>. Citado na página 19.

CRUMPLER, W.; LEWIS, J. A. The Cybersecurity Workforce Gap. p. 10, 2022. Citado na página 19.

(DAI) DECANATO DE PLANEJAMENTO, O. e. A. I. D. Diretoria de Avaliação e I. G. *ANUÁRIO ESTATÍSTICO 2022*. 2022. Disponível em: <https://dpo.unb.br/images/phocadownload/unbemnumeros/anuarioestatistico/Anurio_Estatstico_2022.pdf>. Citado na página 106.

DEMARTINI, F. Golpes no whatsapp: saiba como se proteger dos ataques mais comuns. 2022. Disponível em: <https://canaltech.com.br/seguranca/golpes-no-whatsapp-saiba-como-se-proteger-dos-ataques-mais-comuns-219801/>. Citado 2 vezes nas páginas 19 e 20.

EDUCATION, T. H. *World University Rankings*. 2023. Disponível em: <https://www.timeshighereducation.com/world-university-rankings/2024/world-ranking>. Citado 3 vezes nas páginas 43, 46 e 47.

EDURANK. *World's best Software Engineering universities [Rankings]*. 2021. Disponível em: <https://edurank.org/cs/software-engineering/>. Citado 3 vezes nas páginas 43, 46 e 47.

FUTCHER, L.; SOLMS, R. von. Guidelines for secure software development. In: *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries riding the wave of technology - SAICSIT '08*. Wilderness, South Africa: ACM Press, 2008. p. 56–65. ISBN 978-1-60558-286-3. Disponível em: <http://portal.acm.org/citation.cfm?doid=1456659.1456667>. Citado na página 25.

GATES, B. *Bill Gates: Trustworthy Computing | WIRED*. 2002. Disponível em: <https://www.wired.com/2002/01/bill-gates-trustworthy-computing/>. Citado na página 26.

GAñáN, C. H.; CIERE, M.; EETEN, M. van. Beyond the pretty penny: the Economic Impact of Cybercrime. In: *Proceedings of the 2017 New Security Paradigms Workshop*. Santa Cruz CA USA: ACM, 2017. p. 35–45. ISBN 978-1-4503-6384-6. Disponível em: <https://dl.acm.org/doi/10.1145/3171533.3171535>. Citado na página 19.

GIL, A. C. *Como elaborar projetos de pesquisa*. [S.l.]: Atlas, 2008. Citado na página 35.

GOV.BR. Segurança cibernética. 2022. Disponível em: <https://www.gov.br/anatel/pt-br/assuntos/seguranca-cibernetica>. Citado na página 20.

HAGGAG, O. et al. COVID-19 vs Social Media Apps: Does Privacy Really Matter? In: *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. [S.l.: s.n.], 2021. p. 48–57. Citado na página 19.

HONEYCUTT, J. M.; CANTRILL, J. G. *Cognition, Communication, and Romantic Relationships*. New York: Routledge, 2000. ISBN 978-1-4106-0048-6. Citado na página 19.

HOWARD, M.; LEBLANC, D. *Writing secure code*. 2nd ed. ed. Redmond, Wash: Microsoft Press, 2003. ISBN 978-0-7356-1722-3. Citado na página 26.

IEE; ACM. Software Engineering 2014: Curriculum guidelines for undergraduate degree programs in software engineering. p. 25, 2015. Citado 4 vezes nas páginas 30, 32, 50 e 53.

INDúSTRIA, P. da. O que é Cibersegurança e ameaças cibernéticas? *Portal da Indústria*, 2020. Disponível em: <https://www.portaldaindustria.com.br/industria-de-a-z/ciberseguranca/>. Citado na página 20.

INTERPOL. *INTERPOL report shows alarming rate of cyberattacks during COVID-19*. 2020. Disponível em: <https://www.interpol.int/News-and-Events/News/2020/INTERPOL-report-shows-alarming-rate-of-cyberattacks-during-COVID-19>. Citado na página 19.

ISC. *CYBERSECURITY WORKFORCE STUDY, 2021*. 2021. Disponível em: <https://www.isc2.org/Research/Workforce-Study>. Citado na página 20.

ISO/IEC. *ISO/IEC 27000:2018*. 2018. Disponível em: <https://www.iso.org/standard/73906.html>. Citado na página 23.

IWENDI, C. et al. KeySplitWatermark: Zero Watermarking Algorithm for Software Protection Against Cyber-Attacks. *IEEE Access*, v. 8, p. 72650–72660, 2020. ISSN 2169-3536. Conference Name: IEEE Access. Citado na página 24.

JAHN, S.; MOTTOK, J. Work in Progress: Towards an Academic Secure Software Engineering Curriculum for Engineers. In: *2020 IEEE Global Engineering Education Conference (EDUCON)*. [S.l.: s.n.], 2020. p. 1713–1717. ISSN: 2165-9567. Citado na página 30.

JENSEN, C.; POTTS, C. Privacy Policies as Decision-Making Tools: An Evaluation of Online Privacy Notices. v. 6, n. 1, p. 8, 2004. Citado na página 19.

JR, F. J. F. *Survey research methods*. [S.l.]: Sage publications, 2013. Citado na página 35.

KAZMAN, R.; PASQUALE, L. Software Engineering in Society. *IEEE Software*, v. 37, n. 1, p. 7–9, jan. 2020. ISSN 1937-4194. Conference Name: IEEE Software. Citado na página 19.

KHETARPAL, S. *Post-COVID, 75% of 4.5 lakh TCS employees to permanently work from home by '25; from 20%*. 2020. Disponível em: <https://www.businesstoday.in/latest/corporate/story/post-coronavirus-75-percent-of-3-5-lakh-tcs-employees-permanently-work-from-home-up-from-20-pe Citado na página 19.

LODGHER, A.; YANG, J. Cyber security modules for core, major and elective courses in the bachelor of science (bs) computer science curriculum. *National Security Agency (NSA) Grant*, 2017. Citado na página 29.

LODGHER, A.; YANG, J.; BULUT, U. An Innovative Modular Approach of Teaching Cyber Security across Computing Curricula. In: *2018 IEEE Frontiers in Education Conference (FIE)*. [S.l.: s.n.], 2018. p. 1–5. ISSN: 2377-634X. Citado na página 30.

MASON, R. Four Ethical Issues of the Information Age. *Management Information Systems Quarterly - MISQ*, v. 10, mar. 1986. Citado na página 19.

MCDONALD, S.; TOWEY, D.; BRUSIC, V. Social Impact of Smart Environments: Software Engineering Perspectives and Challenges. In: *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*. [S.l.: s.n.], 2022. p. 1592–1597. ISSN: 0730-3157. Citado na página 19.

MCGRAW, G. Testing for security during development: why we should scrap penetrate-and-patch. *IEEE Aerospace and Electronic Systems Magazine*, v. 13, n. 4, p. 13–15, abr. 1998. ISSN 1557-959X. Conference Name: IEEE Aerospace and Electronic Systems Magazine. Citado na página 26.

MCGRAW, G. *Seven Touchpoints for Software Security*. 2006. Disponível em: <http://www.swsec.com/resources/touchpoints/>. Citado na página 28.

MCLEAN, M. *2022 Must-Know Cyber Attack Statistics and Trends | Embroker*. 2019. Section: Business Advice & Research. Disponível em: <https://www.embroker.com/blog/cyber-attack-statistics/>. Citado na página 19.

MEC. *Diretrizes Curriculares - Cursos de Graduação*. 2016. Disponível em: <http://portal.mec.gov.br/component/content/article?id=12991>. Citado na página 29.

MICROSOFT. *Microsoft Security Development Lifecycle Practices*. 2022. Disponível em: <https://www.microsoft.com/en-us/securityengineering/sdl/practices>. Citado na página 27.

MICROSOFT. *Microsoft Forms*. 2023. Disponível em: <https://forms.office.com/Pages/DesignPageV2.aspx>. Citado na página 56.

MICROSOFT. *Microsoft Security Development Lifecycle*. 2023. Disponível em: <https://www.microsoft.com/en-us/securityengineering/sdl>. Citado na página 28.

MINAYO, M. C. d. S. Análise qualitativa: teoria, passos e fidedignidade. *Ciência & Saúde Coletiva*, v. 17, n. 3, p. 621–626, mar. 2012. ISSN 1413-8123. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-81232012000300007&lng=pt&tlng=pt>. Citado na página 35.

MOLLéRI, J. S.; PETERSEN, K.; MENDES, E. An Empirically Evaluated Checklist for Surveys in Software Engineering. *arXiv:1901.09850 [cs]*, jan. 2019. ArXiv: 1901.09850. Disponível em: <http://arxiv.org/abs/1901.09850>. Citado 2 vezes nas páginas 39 e 105.

NEWHOUSE, W. et al. *National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework*. Gaithersburg, MD, 2017. NIST SP 800–181 p. Disponível em: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-181.pdf>. Citado 3 vezes nas páginas 29, 30 e 33.

OFFICE, C. et al. National Cyber Security Strategy 2016-2021. p. 80, 2017. Citado na página 23.

OWASP. *CLASP*. 2006. Disponível em: <https://owasp.org/www-pdf-archive/Us_owasp-clasp-v12-for-print-lulu.pdf>. Citado na página 28.

PFLEEGER, S.; KITCHENHAM, B. *Principles of survey research: part 1: turning lemons into lemonade. ACM SIGSOFT Softw Eng Notes 26 (6): 16–18*. 2001. Citado na página 35.

RASHID, A. et al. The Cyber Security Body of Knowledge. p. 1067, 2019. Citado 6 vezes nas páginas 20, 23, 24, 25, 26 e 48.

SOFTWARE Engineering in Civic Tech A Case Study about Code for Ireland. In: . [S.l.: s.n.], 2019. p. 41–50. Citado na página 19.

STATISTICS, O. of O.; PROJECTIONS, E. *Field of degree: Computer and information technology : Occupational Outlook Handbook: : U.S. Bureau of Labor Statistics*. 2021. Disponível em: <https://www.bls.gov/ooh/field-of-degree/computer-and-information/computer-and-information-technology-field-of-degree.htm>. Citado na página 47.

STATISTICS, U. B. O. L. *Software Developers, Quality Assurance Analysts, and Testers : Occupational Outlook Handbook: : U.S. Bureau of Labor Statistics*. 2023. Disponível em: <https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm#tab-1>. Citado na página 47.

USNEWS. *The Best Universities in the World, Ranked*. 2023. Disponível em: <https://www.usnews.com/education/best-global-universities/rankings>. Citado 3 vezes nas páginas 43, 46 e 47.

VIEGA, J.; MCGRAW, G. *Building Secure Software*. [s.n.], 2002. ISBN 978-0-672-33409-2. Disponível em: <https://learning.oreilly.com/library/view/building-secure-software/9780672334092/fm.html>. Citado 2 vezes nas páginas 25 e 26.

WOHLIN, C. et al. *Experimentation in software engineering*. [S.l.]: Springer Science & Business Media, 2012. Citado na página 35.

YANG, J.; LODGHER, A.; LEE, Y. Secure Modules for Undergraduate Software Engineering Courses. In: *2018 IEEE Frontiers in Education Conference (FIE)*. [S.l.: s.n.], 2018. p. 1–5. ISSN: 2377-634X. Citado 5 vezes nas páginas 29, 32, 33, 50 e 53.

YASIN, A. et al. Improving software security awareness using a serious game. *IET Software*, v. 13, n. 2, p. 159–169, 2019. ISSN 1751-8814. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/iet-sen.2018.5095. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1049/iet-sen.2018.5095>. Citado na página 19.

YUAN, X.; YANG, L. Secure Software Engineering Education: Knowledge Area, Curriculum and Resources. p. 25, 2016. Citado 2 vezes nas páginas 29 e 30.

ZATKO, S. Rethinking the Role of Security in Undergraduate Education. *IEEE Security & Privacy*, v. 14, n. 2, p. 73–78, mar. 2016. ISSN 1558-4046. Conference Name: IEEE Security & Privacy. Citado 2 vezes nas páginas 39 e 105.

ZHU, J.; LIPFORD, H. R.; CHU, B. Interactive support for secure programming education. In: *Proceeding of the 44th ACM technical symposium on Computer science education - SIGCSE '13*. Denver, Colorado, USA: ACM Press, 2013. p. 687. ISBN 978-1-4503-1868-6. Disponível em: <http://dl.acm.org/citation.cfm?doid=2445196.2445396>. Citado na página 29.

# Apêndices

# APÊNDICE A – Universidades Brasileiras

Esta seção traz o detalhamento dos dados levantados e demonstrados no Capítulo 4, a partir da pesquisa realizada nos currículos do curso de Engenharia de software oferecidos por universidades públicas brasileiras.

## A.1 UnB

Disciplinas ofertadas:

- Criptografia e Segurança de Redes

Conteúdo difuso:

- Fundamentos de sistemas operacionais: **06-Segurança e proteção**
  - Criptografia
  - autenticação de usuário
  - ataques
  - Mecanismos de proteção
- Técnicas de Programação em plataformas emergentes: **01-Programação defensiva**
  - Conceitualização e importância
  - Principais técnicas de programação defensiva
- Fundamentos de redes de computadores: **06-Segurança em redes de computadores**
  - Conceituação
  - Princípios dde criptografia(simétrica e assimétrica)
  - Ataques e contramedidas
  - Controle de acesso e VPNs
  - Aspectos de segurança do desenvolvimento de software (protocolos (HTTPS, SSL/TLS), autenticação (certificados digitais), integridade(assinatura digital))

## A.2 UFC

Disciplinas ofertadas:

- Auditoria e segurança de Sistemas de informação
  - Conceitos e tipos de ameça, riscos e vulnerabilidades dos sistemas de informação.
  - O conceito e objetivos de segurança de informações.
  - O planejamento, implementação e avaliação de políticas de segurança de informações.
  - O conceito e objetivos da auditoria de sistemas de informação.
  - Técnicas de auditoria em sistemas de informação.
  - Softwares de auditoria.
  - Estrutura da função de auditoria de sistemas de informação nas organizações.
- Segurança
  - Ameças.
  - Segurança como atributo qualitativo de projeto de software.
  - Autenticação.
  - Integridade.
  - Confidencialidade.
  - Criptografia(chaves simétricas e assimétricas).
  - Infraestrutura de chaves públicas brasileiras(ICP-Brasil).
  - Certificados digitais.
  - Assinaturas digitais.
  - Desenvolvimento de software seguro.
  - Noções de auditoria de sistemas.
  - Norma NBR 27002.

Conteúdo difuso: Nenhum dado encontrado.

## A.3 UFG

Disciplinas ofertadas: Nenhum dado encontrado.

Conteúdo difuso:

- Construção de Software
  - Fundamentos de codificação (32h): estratégias recomendadas para criar código, variáveis, classes, interfaces, polimorfismo, rotinas, recursão, condições, laços, tratamento de exceção, reflexão, programação defensiva, padrão de codificação (leiaute e estilo), documentação, ferramentas de programação.
- Projeto de Software
  - Questões básicas de design de software: concorrência, controle e tratamento de eventos, persistência de dados, distribuição, tratamento de erro e exceção, tolerância a falhas, interação e apresentação, e segurança.
- Governança e gestão de serviços de software
  - Gestão de riscos de software; riscos organizacionais; riscos relacionados à segurança física e lógica de software.
  - Segurança da informação em uma organização
  - Segurança de software.
- Software para sistemas ubíquos
  - Sistemas de informação que fazem uso de dispositivos (ubíquos) (16h): smartphones, sensores, internet das coisas (IoT), stream analytics e aspectos de segurança (vulnerabilidades, criptografia, certificados digitais).

## A.4 UEPA

Disciplinas ofertadas:

- Projetos de Redes de Segurança
  - Introdução à segurança de sistemas: Princípios da segurança, Ameaças, Vulnerabilidades e Ataques.
  - Normas de segurança da informação (ABNT).
  - Políticas de segurança.
  - Firewall.
  - DNS para redes seguras.
  - IPSec (Internet Protocol Security).
  - Criptografia.
  - Notificação de Incidentes.

  - IDS (IntrusionDetection System).
  - Auditoria e Computação forense.

Conteúdo difuso:

- Fundamentos de sistemas operacionais
  - Segurança e proteção em sistemas operacionais.
- Fundamentos de Redes de Computadores
  - Segurança e autenticação.
- Tópicos especiais em Redes de Computadores
  - Apresenta sempre temas relevantes e atuais sobre o contexto de segurança de redes de computadores. Com estrutura mais flexível, podendo ter sua totalidade em atividades práticas.

## A.5 UFMS

Disciplinas ofertadas:

- Análise Forense Computacional
  - Conceitos básicos análise forense.
  - Procedimentos e Políticas de Segurança.
  - Detecção e identificação de comprometimento da segurança (ataques, identificação da autoria).
  - Coleta e análise de evidências.
  - Reconstrução cronológica do ataque.
  - Técnicas e ferramentas. Recuperação do Sistema.
  - Medidas preventivas.
  - Leis.
- Confiabilidade em Sistemas de Software
  - Visão geral de segurança da informação (Integridade, Confidencialidade e Disponibilidade).
  - Introdução a sistemas confiáveis e seguros.
  - Princípios de design de softwares seguros.

  - Dependability: conceitos, métricas, escopo, análise (incluindo técnicas e ferramentas disponíveis).
  - Atributos e meios para alcançar Dependability.
  - Técnicas de tolerância a falhas.
  - Confiabilidade de Sistemas Dinâmicos.
  - Manutenibilidade e Suporte a Manutenção.
  - Modelagem e Simulação da Dependabilidade de Sistemas.

- Introdução a Criptografia Computacional
  - Requisitos da segurança da informação.
  - Métodos clássicos de ciframento.
  - Criptoanálise elementar.
  - Cifras de bloco versus cifras de fluxo.
  - Técnicas para ciframento encadeado.
  - Fundamentos matemáticos da criptografia moderna.
  - Técnicas básicas para a geração de números pseudo- aleatórios.
  - Algoritmos modernos de ciframento: simétricos ou de chave secreta, assimétricos ou de chave pública.
  - Assinaturas digitais: algoritmos e protocolos para autenticação de usuários e não repúdio de envio de mensagens.

- Segurança de Redes
  - Segurança da informação.
  - Padrões de Segurança e a ISO.
  - Classificação da informação.
  - Vulnerabilidades e ataques.
  - Autenticação. Criptografia, assinatura digital, técnicas de cifragem.
  - Mecanismos e ferramentas de segurança.

- Segurança e Auditoria de Sistema
  - Conceitos Básicos de Segurança da Informação.
  - Classificação da Informação.
  - Riscos e Impactos (Zonas de Segurança).
  - Topologias Seguras.

  - Controle de Acesso.
  - O planejamento, implementação e avaliação de políticas de segurança de informações.
  - Vulnerabilidades e Ameaças.
  - O conceito e os objetivos da auditoria de sistemas de informação.
  - Pontos de Controles de Auditoria de Sistemas.
  - Plano de Contingência e Continuidade dos Serviços.
  - Técnicas de auditoria em sistemas de informação. Estudos de Caso.

Conteúdo difuso:

- Computação Distribuída
  - Conceitos básicos: arquiteturas, processos, comunicação, nomeação, sincronização, consistência e replicação, tolerância a falhas, **segurança**.
- Computação e Sociedade
  - Segurança, privacidade, direitos de propriedade, acesso não autorizado.
  - Crimes de informática.
  - Computação forense e Direito da Informática.
- Engenharia de requisitos
  - Requisitos de Segurança.
- Fundamentos de Redes de Computadores
  - Noções de segurança.
- Introdução a Sistemas Operacionais
  - Noções de Segurança.
- Programação para Dispositivos Móveis
  - Fundamentos de Segurança para aplicativos.
- Redes de Computadores
  - Noções de segurança e autenticação.
- Redes sem fio
  - Segurança e autenticação em redes sem fio.

- Sistemas Distribuídos
  - Segurança.

## A.6 UFAM

Disciplinas ofertadas: Nenhum dado encontrado.

Conteúdo difuso:

- Engenharia de Requisitos
  - Fundamentos (completitude, consistência, robustez, análise estática, simulação, verificação de modelos, **segurança, safety**, usabilidade, desempenho, análise de causa/ efeito, priorização, análise de impacto e rastreabilidade).
- Engenharia de Software II
  - Questões básicas de design de software: concorrência, controle e tratamento de eventos, persistência de dados, distribuição, tratamento de erro e exceção, tolerância a falhas, interação e apresentação, e **segurança**.
- Programação para Computação Móvel
  - **Segurança** das aplicações móveis.
- Redes de Computadores
  - Tópicos avançados em redes (redes sem fio, p2p, multimídia e **segurança**).
- Teste, Verificação e Validação de Software
  - Aplicação prática de teste de unidade, teste de sistemas, testes funcionais, teste de configuração, teste de integridade, teste de integração, teste de aceitação, teste de regressão, **teste de segurança**, teste de performance e teste de volume.
- Engenharia de Aplicações WEB
  - Segurança.
- Projeto Detalhado de Software
  - Atributos qualitativos em um projeto (confiabilidade, usabilidade, manutenibilidade, testabilidade, desempenho, **segurança**, tolerância a falhas e outros).
- Técnicas de Testes

  - Conceito sobre processo de teste. Tipos de teste: caixa-preta, caixa-branca, regressão, carga, estresse, usabilidade e segurança.

## A.7 UFRN(a distancia)

Disciplinas ofertadas: Nenhum dado encontrado.

Conteúdo difuso:

- Boas Práticas de programação
  - Programação defensiva.

## A.8 UDESC

Disciplinas ofertadas: Nenhum dado encontrado.

Conteúdo difuso:

- Sistemas Operacionais
  - Segurança e proteção
- Persistência de Dados
  - Segurança com restrições de acesso.
- Redes de Computadores
  - Segurança e autenticação.
- Desenvolvimento de Sistemas Críticos
  - Segurança funcional crítica.
- Ética, computador e Sociedade.
  - Privacidade.

## A.9 UFTPR

Disciplinas ofertadas:

- Segurança e Auditoria em Sistemas
  - Auditoria de sistemas.

  - Segurança de sistemas.
  - Metodologia de auditoria.
  - Análise de riscos.
  - Plano de contingência.
  - Técnicas de avaliação.
  - Aspectos especiais: vírus, fraudes, criptografia, acesso não autorizado.

  Conteúdo difuso:

- Sistemas Operacionais
  - Noções de proteção e segurança.

## A.10 UFERSA

Disciplinas ofertadas:

- Dependabilidade e Segurança
  - Tolerância a falhas: definição, redundância de hardware e de software, algoritmos tolerantes a falhas e técnicas de projeto de sistemas tolerantes a falhas.
  - Segurança: conceitos básicos, criptografia e tipos de criptografia, mecanismos de proteção e de autenticação, tipos de ataques, malwares e defesas.

  Conteúdo difuso: Nenhum dado encontrado.

## A.11 UNIPAMPA

Disciplinas ofertadas:

- Tópicos de Segurança de Sistemas e da Informação
  - Introdução à Segurança.
  - Conceitos Básicos.
  - Primitivas Criptográficas.
  - Criptografia Simétrica e de Chave Pública.
  - Assinatura e Certificação Digital.
  - Propriedades de Segurança.

  - Protocolos de Autenticação.
  - Tecnologias de Segurança de Sistemas e Dados.

- Tópicos em Ataque e Defesa de Sistemas
  - Conceitos básicos.
  - Técnicas que precedem um ataque.
  - Metodologias, técnicas e ferramentas para realização de ataques.
  - Falhas de software e hardware que podem levar ao sucesso de um ataque.
  - Técnicas e tecnologias atuais que ajudam a mitigar os efeitos de um ataque.

Conteúdo difuso:

- Tópicos em Resolução de Problemas em Sistemas Unix/Linux II
  - Gerenciamento, escalabilidade, balanceamento de carga e **segurança de servidores** essenciais da Internet, como DNS, Web e bancos de dados.

# APÊNDICE B – Universidades estrangeiras (EUA)

Esta seção traz o detalhamento dos dados levantados e demonstrados no Capítulo 4, a partir da pesquisa realizada nos currículos do curso de Ciência da computação por universidades referência nos EUA.

## B.1 Berkley

Disciplinas ofertadas:

- COMPSCI 161 Computer Security
  - Introduction to computer security;
  - Cryptography, including encryption, authentication, hash functions, cryptographic protocols, and applications;
  - Operating system security, access control;
  - Network security, firewalls, viruses, and worms;
  - Software security, defensive programming, and language-based security;
  - Case studies from real-world systems.
- COMPSCI 171 Cryptography
  - Mathematical definition of secure encryption;
  - Secure encryption construction, assuming computational hardness.
- COMPSCI 261 Security in Computer Systems
  - Graduate survey of modern topics in computer security;
  - Protection;
  - Access control;
  - Distributed access security;
  - Firewalls;
  - Secure coding practices;
  - Safe languages;

  - Mobile code;
  - Case studies from real-world systems;
  - May also cover cryptographic protocols, privacy and anonymity, and/or other topics as time permits.

- COMPSCI 261N Internet and Network Security
  - Denial-of-service;
  - Capabilities;
  - Network intrusion detection/prevention;
  - Worms;
  - Forensics;
  - Scanning;
  - Traffic analysis;
  - Legal issues;
  - Web attacks;
  - Anonymity;
  - Wireless and networked devices;
  - Honeypots;
  - Botnets;
  - Scams;
  - Underground economy;
  - Attacker infrastructure;
  - Research pitfalls.

Conteúdo difuso:

- COMPSCI C8 Foundations of Data Science
  - It delves into social and legal issues surrounding data analysis, including issues of privacy and data ownership.
- COMPSCI 10 The Beauty and Joy of Computing
  - Social implications of computing (privacy, education, algorithmic bias).
- COMPSCI 162 Operating Systems and System Programming
  - Protection, security, and privacy.

- COMPSCI 169A Introduction to Software Engineering
  - Practical performance and security in software operations.
- COMPSCI 195 Social Implications of Computer Technology
  - Intellectual property;
  - Privacy;
- COMPSCI H195 Honors Social Implications of Computer Technology
  - Intellectual property;
  - Privacy;
- COMPSCI 260B Human-Computer Interaction Research
  - Usable security.
- COMPSCI 262A Advanced Topics in Computer Systems
  - security infrastructure.

## B.2 Stanford

Disciplinas ofertadas:

- CS153 - COMPUTER SECURITY AT SCALE
  - Confidential Computing;
  - Privacy;
  - Trust;
  - Safety and Real World.
- CS155 - COMPUTER AND NETWORK SECURITY
  - Principles of computer systems security.
  - Attack techniques and how to defend against them.
  - Topics include: network attacks and defenses, operating system security, application security (web, apps, databases), malware, privacy, and security for mobile devices.
  - Course projects focus on building reliable so6ware.
- CS251 - CRYPTOCURR AND BLOCKCHAIN TECH

  - Students will learn how these systems work, and how to engineer secure software that interacts with Blockchains like Bitcoin, Ethereum, and others.

- CS253 - WEB SECURITY
  - Principles of web security;
  - The fundamentals and state-of-the-art in web security;
  - Attacks and countermeasures;
  - Topics include: the browser security model, web app vulnerabilities, injection, denial-of-service, TLS attacks, privacy, fingerprinting, same-origin policy, cross site scripting, authentication, JavaScript security, emerging threats, defense-in-depth, and techniques for writing secure code.
  - Writing security exploits;
  - Defending insecure web apps;
  - Implementing emerging web standards.

- CS255 - INTRO CRYPTOGRAPHY
  - Encryption (symmetric and public key);
  - Digital signatures;
  - Data integrity;
  - Authentication;
  - Key management;
  - PKI;
  - Zero-knowledge protocols;
  - Real-world applications.

- CS294S - RES PROJ IN SOFTWR SYS & SEC
  - Programmable open mobile Internet;
  - Human-computer interaction;
  - Programming systems;
  - Operating systems;
  - Networking;
  - Security.

- CS350 - SECURE COMPILATION
  - Threat models for secure compilers;

  - Formal criteria for secure compilers to adhere to;
  - Security relevance of secure compilation criteria;
  - Security architectures employed to achieve secure compilation;
  - Proof techniques for secure compilation with a focus on backtranslation.

- CS356 - TOPICS COMP NETWORK SECURITY
  - No topics specified.

Conteúdo difuso:

- CS104 - INTRO TO SOFTWARE SYS & TOOLS
  - We propose to develop a course that will teach students the skills necessary to be successful computer scientists, such as the command line, source code management and debugging, security and cryptography, containers and virtual machines, and cloud computing
  - With this deeper understanding, students can leverage critical thinking skills to intelligently and efficiently configure and troubleshoot so6ware systems, assess the security and efficiency of particular tool usages, and synthesize new automation pipelines that integrate multiple tools.
- CS106E - EXPLORATION OF COMPUTING
  - We will then use our foundation to explore a variety of tech-related topics including Computer Security (how computers are attacked and defensive measures that can be taken);
- CS106S - CODING FOR SOCIAL GOOD
  - Principles of cybersecurity.
- CS142 - WEB APPLICATIONS
  - Issues in web security and application scalability.
- CS196 - COMPUTER CONSULTING
  - Topics include operating systems, networking, **security**, troubleshooting methodology with emphasis on Stanford's computing environment.
- CS208E - GREAT IDEAS IN CS
  - Cryptography and security;

- CS240 - ADV TOPICS OPERATING SYSTEMS
  - Protection and security;
- CS243 - PROGRAM ANALYS & OPTIMIZ
  - Program analysis techniques used in compilers and so6ware development tools to improve productivity, reliability, and **security**.
- CS249I - THE MODERN INTERNET
  - Pressing privacy, security, and abuse challenges.
- CS329S - MACHINE LEARNING SYS DESIGN
  - In the process, students will learn about important issues including privacy, fairness, and security
- CS349D - CLOUD COMPUTING TECHNOLOGY
  - Programming interfaces;
  - Cloud native applications;
  - Resource management;
  - Pricing;
  - Availability;
  - Reliability;
  - Privacy;
  - Security.

## B.3 Carnegie Mellon

Disciplinas ofertadas:

- 15-316 Software Foundations of Security and Privacy
  - Policy models and mechanisms for confidentiality;
  - Integrity;
  - Availability;
  - Language-based techniques for detecting and preventing security threats;
  - Mechanisms for enforcing privacy guarantees;

  - Interaction between software and underlying systems that can give rise to practical security threats.

- 15-330 Introduction to Computer Security

  - Students will learn the basic concepts in computer security including software vulnerability analysis and defense, networking and wireless security, and applied cryptography.
  - Students will also learn the fundamental methodology for how to design and analyze security critical systems.

- 15-356 Introduction to Cryptography

  - We will cover formal definitions of security, as well as constructions based on well established assumptions like factoring.

- 15-392 Special Topic: Secure Programming

  - This course provides a detailed explanation of common programming errors in C and C++ and describes how these errors can lead to software systems that are vulnerable to exploitation.
  - The course concentrates on security issues intrinsic to the C and C++ programming languages and associated libraries.
  - It does not emphasize security issues involving interactions with external systems such as databases and web servers, as these are rich topics on their own.
  - Topics to be covered include the secure and insecure use of integers, arrays, strings, dynamic memory, formatted input/output functions, and file I/O.

Conteúdo difuso:

- 15-346 Computer Architecture: Design and Simulation

  - Several frontiers of current research will be explored in energy efficiency and security threats.

## B.4 Massachusetts Institute of Technology

Disciplinas ofertadas:

- 6.1600 Foundations of Computer Security

  - Cryptographic foundations (pseudorandomness, collision-resistant hash functions, authentication codes, signatures, authenticated encryption, public-key encryption);
  - Systems ideas (isolation, non-interference, authentication, access control, delegation, trust);
  - Implementation techniques (privilege separation, fuzzing, symbolic execution, runtime defenses, side-channel attacks).
  - Case studies of how these ideas are realized in deployed systems.
  - Lab assignments apply ideas from lectures to learn how to build secure systems and how they can be attacked.

  Conteúdo difuso:

- 6.1800 Computer Systems Engineering
  - Security and privacy;

## B.5 Harvard

Disciplinas ofertadas:

- COMPSCI 105 - Privacy and Technology
  - This course critically examines popular concepts of privacy and uses a rigorous analysis of echnologies to understand the policy and ethical issues at play.
  - Case studies: database anonymity, research ethics, wiretapping, surveillance, and others. Course relies on some technical material, but is open and accessible to all students, especially those with interest in economics, engineering, political science, computer science, sociology, biology, law, government, philosophy.
- COMPSCI 263 - Systems Security
  - Buffer overflows;
  - Web security;
  - Information flow control;
  - Anonymous communication mechanisms suchas Tor.
- COMPSCI 362 - Software Systems: Security, Performance, and Robustness
  - No info.

- COMPSCI 364 - Programming Languages and Security
  - No info.

  Conteúdo difuso: nenhum dado encontrado.

## B.6 California Institute of Technology - Caltech

Disciplinas ofertadas: Nenhum dado encontrado

Conteúdo difuso:

- CS 132. Web Development
  - Concepts including separation of concerns, the client-server relationship, user experience, accessibility, and **security** will also be emphasized throughout the course.

## B.7 Princeton

Disciplinas ofertadas:

- COS432 - Information Security
  - Security issues in computing, communications, and electronic commerce.
  - Goals and vulnerabilities;]
  - Legal and ethical issues;
  - Basic cryptology;
  - Private and authenticated communication;
  - Electronic commerce;
  - Software security;
  - Viruses and other malicious code;
  - Operating system protection;
  - Trusted systems design;
  - Network security;
  - Firewalls;
  - Policy, administration and procedures;
  - Auditing;

  - Physical security;
  - Disaster recovery;
  - Reliability;
  - Content protection;
  - Privacy.

- COS433 - Cryptography
  - Chosen ciphertext security;

Conteúdo difuso:

- COS109 - Computers in Our World
  - How the Internet and the Web work, security and privacy.
- COS316 - Principles of Computer System Design
  - Students will also learn general systems concepts that support design goals of modularity, performance, and **security**.
- COS441 - Programming Languages
  - Safety and security guarantees.
- COS461 - Computer Networks
  - Network security.

## B.8 The University of Chicago - UChicago

Disciplinas ofertadas:

- CMSC 23200. Introduction to Computer Security.
  - This course introduces the principles and practice of computer security.
  - It aims to teach how to model threats to computer systems and how to think like a potential attacker.
  - It presents standard cryptographic functions and protocols and gives an overview of threats and defenses for software, host systems, networks, and the Web.
  - It also touches on some of the legal, policy, and ethical issues surrounding computer security in areas such as privacy, surveillance, and the disclosure of security vulnerabilities.

  - The goal of this course is to provide a foundation for further study in computer security and to help better understand how to design, build, and use computer systems more securely.

- CMSC 23206. Security, Privacy, and Consumer Protection.
  - Basic cryptography; physical, network, endpoint, and data security;
  - Privacy (including user surveillance and tracking); attacks and defenses; and relevant concepts in usable security.
  - Consumer privacy;
  - Censorship;
  - Platform content moderation;
  - Data breaches;
  - Net neutrality;
  - Government surveillance;
  - Election security;
  - Vulnerability discovery and disclosure;
  - The fairness and accountability of automated decision making, including machine learning systems.

- CMSC 23210. Usable Security and Privacy.
  - This course will examine how to design for security and privacy from a user-centered perspective by combining insights from computer systems, human-computer interaction (HCI), and public policy.
  - Core security;
  - Privacy technologies, as well as HCI techniques for conducting robust user studies
  - Usable authentication;
  - User-centered web security;
  - Anonymity software;
  - Privacy notices;
  - Security warnings;
  - Data-driven privacy tools in domains ranging from social media to the Internet of Things.

- CMSC 23218. Surveillance Aesthetics: Provocations About Privacy and Security in the Digital Age.

  - Privacy and security issues at the intersection of the physical and digital worlds;
  - hrough both computer science and studio art, students will design algorithms, implement systems, and create interactive artworks that communicate, provoke, and reframe pervasive issues in modern privacy and security.

- CMSC 25900. Ethics, Fairness, Responsibility, and Privacy in Data Science.
  - The course will demonstrate how computer systems can violate individuals' privacy and agency, impact sub-populations in disparate ways, and harm both society and the environment. It will also introduce algorithmic approaches to fairness, privacy, transparency, and explainability in machine learning systems.

- CMSC 28400. Introduction to Cryptography.
  - Algorithms for symmetric-key and public-key encryption;
  - Authentication;
  - Digital signatures;
  - Hash functions, and other primitives.

Conteúdo difuso:

- CMSC 11900. Introduction to Data Science II.
  - More advanced topics on data privacy and ethics, reproducibility in science, data encryption, and basic machine learning will be introduced.

## B.9 Johns Hopkins University - JHU

Disciplinas ofertadas:

- EN.601.104. Computer Ethics
  - Privacy issues;
  - Computer crime;
  - Intellectual property law – specifically copyright and patent issues;
  - Globalization;
  - Ethical responsibilities for computer science professionals;

- EN.601.340. Web Security
  - Concepts behind Web security, such as same-origin policy, cross-origin resource sharing, and browser sandboxing;

  - Most popular Web vulnerabilities, such as cross-site scripting (XSS) and SQL injection, as well as how to attack and penetrate software with such vulnerabilities;
  - How to detect, respond, and recover from security incidents.

- EN.601.443. Security & Privacy in Computing
  - Computer security;
  - Network security;
  - Basic cryptography;
  - System design methodology;
  - Privacy

- EN.601.444. Network Security
  - Authentication
  - Access control;
  - Integrity and confidentiality of data;
  - Firewalls and related technologies;
  - Web security and privacy.

- EN.601.445. Practical Cryptographic Systems
  - Techniques used in practical security systems;
  - Mistakes that lead to failure;
  - Approaches that might have avoided the problem.
  - Techniques of provable security and the feasibility of reverse-engineering undocumented cryptographic systems.

- EN.650.631. Ethical Hacking.
  - Skills needed to defend computer network infrastructure by exposing them to the hands-on identification and exploitation of vulnerabilities in servers (i.e., Windows and Linux), wireless networks, websites, and cryptologic systems;
  - Shell coding;
  - IDA Pro analysis;
  - Fuzzing;
  - Writing or exploiting network-based applications or techniques such as web servers, spoofing, and denial of service.

- EN.601.740. Language-based Security
  - Control-flow and data-flow graphs;
  - Program slicing;
  - Code property graph (CPG);
  - Control-flow integrity.
- EN.601.743. Advanced Topics in Computer Security.
  - Network perimeter protection;
  - Host-level protection;
  - Authentication technologies;
  - Intellectual property protection;
  - Formal analysis techniques;
  - Intrusion detection and similarly advanced subjects

Conteúdo difuso:

- EN.601.318. Operating Systems.
  - Protection and security;
- EN.601.414. Computer Networks
  - Security issues such as firewalls and denial of service attacks;
  - Security protocols (e.g. TLS, SSH, IPsec), as well as some basic cryptography necessary to understand these.
- EN.601.422. Software Testing & Debugging
  - Security testing;
- EN.601.745. Advanced Topics in Applied Cryptography.

## B.10 University of Washington

Disciplinas ofertadas:

- CSE 426 Cryptography
  - Students learn to formalize security goals, design schemes for achieving these goals, and study security attacks or security proofs that establish the security or insecurity of schemes.

- CSE 484 Computer Security
  - Foundations of modern computer security;
  - Software security;
  - Operating system security;
  - Network security;
  - Applied cryptography;
  - Human factors;
  - authentication;
  - anonymity;
  - Web security.

  Conteúdo difuso:

- CSE 154 Web Programming
  - Web security;
- CSE 454 Advanced Internet and Web Services
  - Construction of scalable and secure web services.
- CSE 461 Introduction to Computer-Communication Networks
  - network security.

# APÊNDICE C – Checklist utilizado para criação do survey

## C.1 Research objectives

1A - Are the research objective expressed in measurable terms? E.g. as research questions, or using the goal-question-metric approach.

1B - Is the research context defined? Does it consider a reasonable set of objectives? **Sim. Busca transparecer a visão de engenheiros de software sobre a característica de segurança, como membro do time desenvolvedor, usuário de aplicações e também durante a sua formação, assim como os tópicos importantes para uma disciplina com foco em segurança**

1C - Is the need for a survey research motivated (i.e. grounded on background and related studies)? **Sim** (ZATKO, 2016)

## C.2 Study plan

2A - Is the survey process conducted based upon detailed procedures? Ideally, the survey process should also be based upon methodological guidelines. **Sim** (MOLLéRI; PETERSEN; MENDES, 2019)

2B - Is there a reflection on the need to update the research plan? E.g. through keeping a research diary or log book. **Sim. A maior procupação inicial é sobre o recrutamento de participantes.**

2C - Are the roles and responsibilities of researchers and other stakeholders defined? This information can be detailed in the research plan (see item 2B). **Sim**

## C.3 Identify population

3A - Is the population or the survey's target audience characterized (e.g. through audience analysis)? **Sim. Os alunos de Engenharia de Software na UnB**

3B - Is the size of the population stated? If it is not possible to gather this data, are statistic estimates of the population drawn? **Sim. População estimada através do Anuário Estatístico da UnB de 2022. Quatidade de alunos registrados no**

**curso é de 1015 no segundo semestre de 2022 ((DAI) DECANATO DE PLANEJAMENTO, 2022)**

## C.4 Sampling plan

4A - Is the kind of sample (i.e. probabilistic, non-probabilistic) defined? Obs. impact for data analysis, its representativeness and/or generalization should be discussed. **Sim. Não probabilistico com amostragem por Auto-seleção.**

4B - Is the sampling process described, and the resulting sample size presented? **Sim. A forma de divulgação escolhida foi de divulgar o questionário em grupos de desenvolvedores, alunos e professores da Engenharia de Software.**

4C - Are the sources of sampling (e.g. particular databases or directories, open or restricted) defined? E.g. through a search plan.

4D - Are the strategies and criteria to select units (of observation, of analysis and search unit) stated? E.g. through a sampling frame.

## C.5 Instrument design

5A - Is the type instrument (i.e. self- or interviewer-administrated) defined? Obs. impact for participant recruitment and manage responses should be discussed. **Sim. Self-administered, com impactor na participação sendo analisados durante o período de divulgação do instrumento. Mais informações na seção 7.**

**5B** - Is the instrument design process (acquisition, development, prototyping, versioning, reuse) described in the report? **Sim**

**5C** - Are the demographic questions formulated according to the audience? If a stratification of the sample is planned, are the demographics adequate to characterize subsets the participants? **Sim**

5D - Has special care been taken to make the questions understandable by the respondents? E.g. through using a terminology familiar to the target population, or by providing a thesaurus. **Sim. O instrumento utiliza de descrições para os tópicos não usuais.**

5E - Has special care been taken to avoid intrusive and unethical questions? E.g. such biases may include questions that lead the respondent to a particular answer, or to expose personal data or behavior. **Sim**

5F - Is the number and order of the questions taken into consideration? In case of a potential bias related to the order of questions is identified, different versions of the

instrument can be distributed. **Sim**

5G - Is there a reflection on the type of responses (i.e. open-ended, close-ended or a mix of both) required for the questions? Ideally, it should be possible to assess the type of each question, but the report could present the overall reasoning for the choices and provide a way to access the instrument.

5H - If employing close-ended questions, are the standardized response formats (i.e. nominal, ordinal, interval or ratio) stated? Appropriate scales should be attributed to the questions according to the mapped variables.

5I - Is there a reflection on the adoption of additional sources for data collection? E.g. through the participant's profile or supporting literature? Such additional sources may provide means for characterizing strata of participants or for validating data through cross-verification and triangulation. **Não. Validação de dados além da execução do survey não é considerada necessária.**

## C.6 Instrument validation

6A - Is the validation process of the survey instrument detailed? E.g. through piloting, pre-test, retest, focus groups, experiments, expert or non expert reviews. The validation should account for issues related to the instrument design (see items 5C-5H). **Sim. A proposta é de aplicar o questionário inicial em um pequeno grupo, com três ou quatro pessoas, visando principalmente observar o tempo de resposta e possíveis dificuldades dos participantes.**

6B - Is the instrument measuring what is intended? Are the questionnaire items mapped to the research question(s)? **Sim**

6C - In the case of an electronic or online questionnaire, is the usability evaluated? E.g. questionnaire navigation, instructions of use, option to resume answering, progress indicator, required/non- inputs, aesthetics, and layout. **Sim. Cuidado foi tomado com o tamanho do questionário, assim como as opções de restrição ao acesso. Aqui a opção foi tomada de limitar os participantes ao curso de engenharia de software com uma pergunta, assim como a restrição de acesso apenas aos emails com o dominio @aluno.unb.br.**

6D - Are the results of the instrument validation discussed? After the main problems been identified, were the instrument updated/amended according to the validation results? **Sim. A validação faz parte do cronograma de execução do trabalho, com uma semana dedicada para a validação e reestruturação(se visto necessário) do instrumento.**

## C.7 Participant recruitment

7A - Are the strategies to select participants (stage 4. Sampling plan) implemented? E.g. through invitations, authorization codes, self-recruitment, or snowballing. **Sim. Principalmente feito através da divulgação aleatória do instrumento, com o possível compartilhamento por participantes.**

7B - Are the ancillary documents (e.g. invitation, cover and thank you letter) provided? If they were not produced, are the reasons for that discussed and convincing?

7C - If rewards or incentives to respondents are provided, are the reasons and implications (e.g. ethical concerns, biases) discussed? Those actions are likely to impact the participant's willing to respond and the research's ethical concerns, thus introducing validity bias. **Não se aplica.**

## C.8 Response management

8A - Are the responses monitored? E.g. response rate, non-responsiveness, and drop-out questions. In case of inadequate response rate, the reasons for non-responses and drop-out items were investigated? **Sim. Processo será monitorado, buscando identificar problemas e solucioná-los para uma possível segunda publicação.**

8B - Is there any action to be taken in case of non-responses (e.g. reminders)? If reminders are employed, is the process for selecting and inviting new participants described? Moreover, are the implications of reminders discussed? I.e. changes in the sample size are likely to impact the heterogeneity and generalizability of data.

## C.9 Data analysis

9A - Is the data validated prior to analysis? E.g. through checking inconsistent, incomplete and missing values. **Sim**

9B - Is the method for data analysis specified? Are the steps of the analysis process described? Are they suitable for the response formats collected?

9C - If statistical analysis is employed, is the hypothesis testing process documented and the standardized responses presented? E.g. through tables, graphs, charts and plots

9D - If using qualitative synthesis (e.g. meta-ethnography, thematic or content analysis), is it clear how the categories/themes were derived from the data?

9E - If a stratified sample is defined (see 5C), are the data analysed according to demographics? Are there meaningful comparisons drawn from them?

## C.10 Reporting

10A - Are the instrument and ancillary documents accessible (e.g. URL link, external reference, appendix) to readers? If not, are the reasons for that discussed and convincing? If data resulting from the survey were disclosure, were anonymity and confidentiality of data discussed?

10B - Has a discussion of both positive and negative findings been demonstrated? Are the discussion addressing the research question(s) or hypothesis? Does the discussion take into consideration the generalization of the findings?

10C - Are the results of the assessment checklist reported? Are limitations of the study (e.g. threats to validity) discussed?

10D - Are the conclusions justified by the results? Furthermore, are the implications and potential use of the results discussed?

# APÊNDICE D – Dados Coletados no questionário

Figura 19 – Q1- Termo de consentimento


Figura 20 – Q2- Você é aluno do curso de Engenharia de Software ?

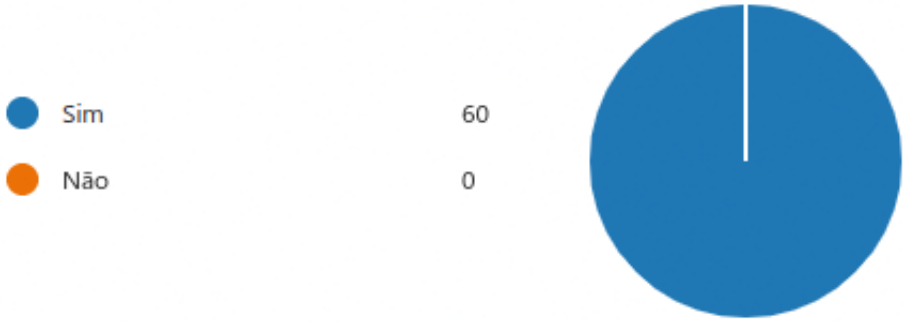



Figura 21 – Q3- Diversos temas são abordados dentro da formação de um Engenheiro de Software. Ordene as áreas de conhecimento abaixo conforme a sua visão de importância para a construção de um produto de software.

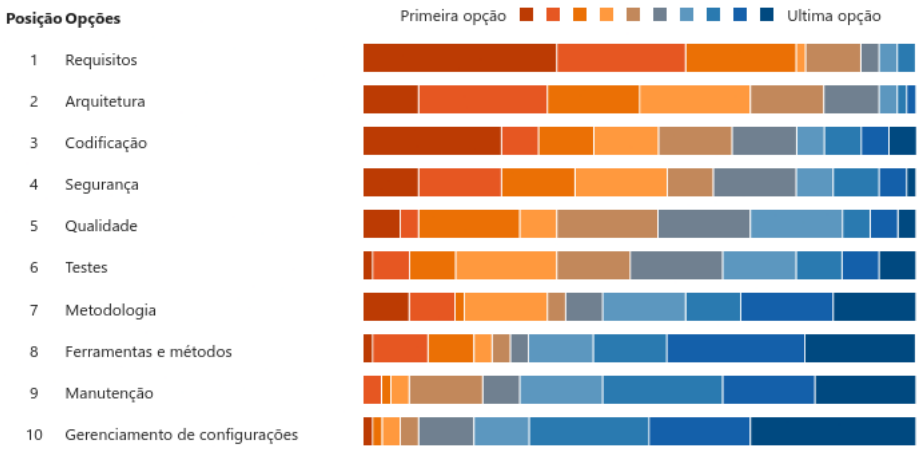


Figura 22 – Questões sobre a importância do tema de segurança de Software.

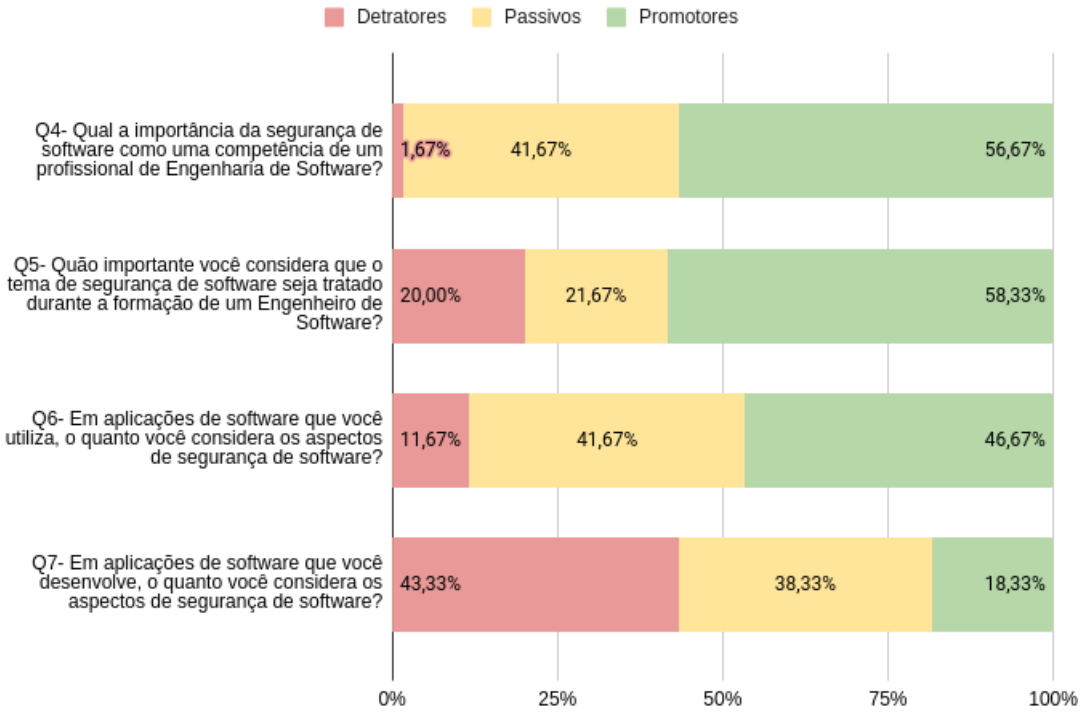



Figura 23 – Q8- Você considera que o currículo do curso de Engenharia de Software na UnB dá atenção adequada para o tema de segurança?


Figura 24 – Q9- Na sua opinião, qual seria a implementação mais adequada para a abordar o tema de segurança no curso de Engenharia de Software?

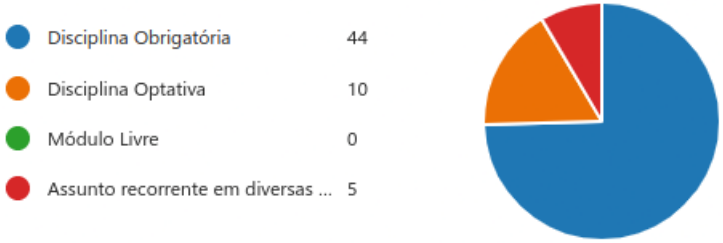


Figura 25 – Q10- Marque os cinco tópicos voltados à segurança de software que você considera mais importantes de serem abordados no curso de Engenharia de Software.


Figura 26 – Q11- Qual o ano que você ingressou no curso de Engenharia de Software?

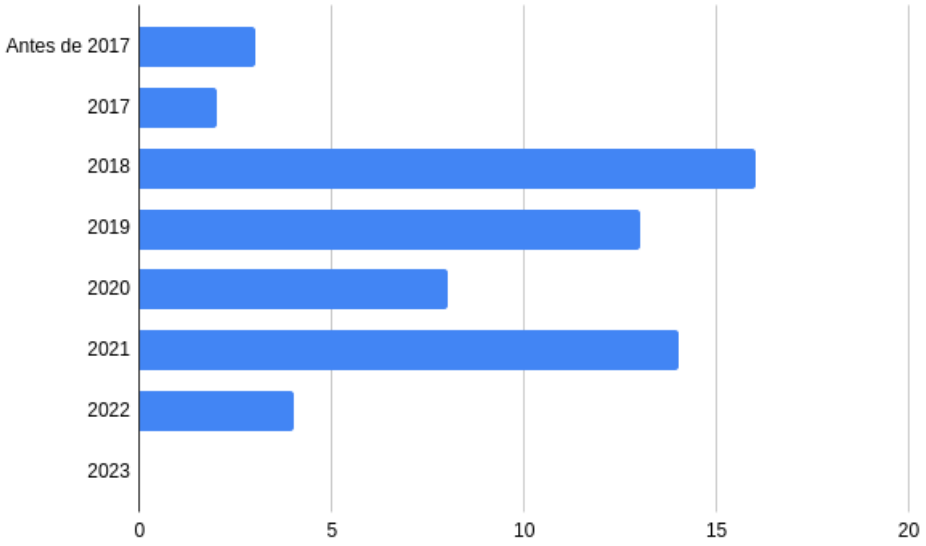



Figura 27 – Q12- Você já atua no mercado de trabalho como engenheiro de software (estágio, CLT, PJ, MEI)?

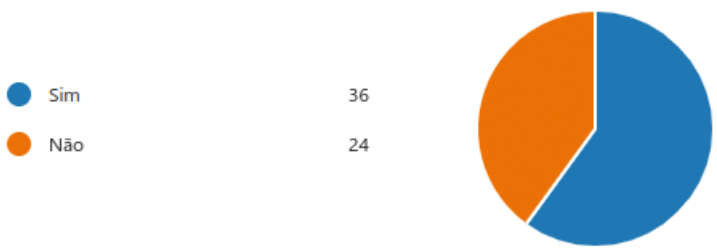


Figura 28 – Q13- Quanto tempo de experiência profissional você possui na área de desenvolvimento de software?

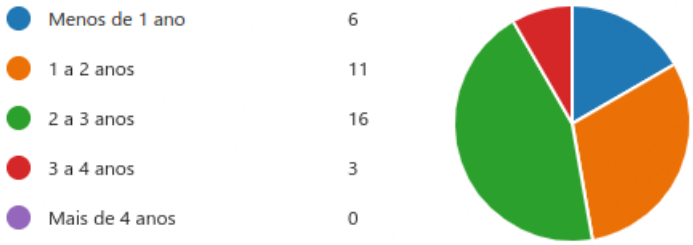



Figura 29 – Q14- Você já aplicou alguma prática de segurança de software no seu trabalho? Por exemplo: modelagem de ameaças, práticas de codificação segura, testes de segurança, etc.

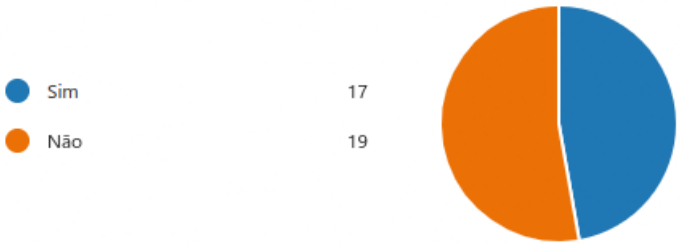



Figura 30 – Q15- Se você for atribuído à alguma tarefa relacionada à segurança da aplicação, você se considera seguro(a) para executá-la?

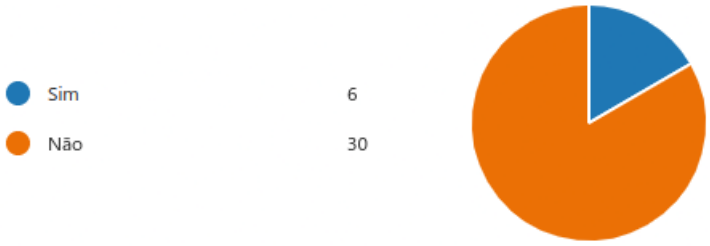



Figura 31 – Q16- Você já recebeu algum tipo de treinamento (fora da universidade) para aplicar práticas de segurança durante o ciclo de desenvolvimento de software?