

Intersecção entre Linhas de Produtos de Software e Sistemas Ciberfísicos: um Mapeamento Sistemático

Luiz Antonio Dassan Neto

Orientador: Prof. Dr. Ricardo Theis Geraldi

Faculdade de Computação (FACOM) – Universidade Federal de Mato Grosso do Sul (UFMS), Campo Grande (MS), Brazil

luiz.dassan@ufms.br, ricardo.geraldi@ufms.br

Resumo. *Linha de Produto de Software (LPS) é um paradigma de desenvolvimento de software que visa o desenvolvimento de sistemas utilizando um conjunto de componentes compartilhados de ativos reutilizáveis, com foco em reutilização de software, redução de custos, melhorar a qualidade e reduzir o tempo de desenvolvimento. Sistemas Ciberfísicos (SCF) é a integração de um sistema computacional com um ambiente físico. Estes sistemas combinam capacidades de comunicação, controle e computação para interagir com o mundo real. A convergência entre LPS e SCF ocorre justamente na necessidade de lidar com a variabilidade e complexidade dos SCF, permitindo que soluções mais flexíveis, reutilizáveis e adaptáveis sejam construídas de forma sistemática. A aplicação de LPS em SCF é promissora e permite lidar com a variabilidade desses sistemas por meio da reutilização sistemática de componentes, promovendo maior eficiência, qualidade e flexibilidade no desenvolvimento. Esta pesquisa tem o objetivo principal de investigar a intersecção de pesquisas entre LPS e SCF por meio do método de pesquisa denominado mapeamento sistemático. O método de mapeamento sistemático foi adotado e organizado em três etapas para recuperar, filtrar e selecionar os estudos primários que fazem parte deste trabalho. Os resultados obtidos a partir da análise desses estudos primários apresentam uma análise detalhada sobre como as LPS têm sido aplicadas e adaptadas no contexto de SCF, com ênfase nas práticas de Gerenciamento de Variabilidades (GV). Este trabalho destaca os frameworks e plataformas mais comuns utilizados para gerenciar a variabilidade em SCF, como o FeatureIDE, SimPL e Engenharia de Sistemas Baseada em Modelos, discutindo sua eficácia para enfrentar os desafios de gerenciar e modelar SCF. Por fim, os resultados oferecem uma visão sobre os desafios contínuos da integração de LPS e GV em ambientes complexos e dinâmicos de SCF, fornecendo uma base para futuras pesquisas e aplicações neste domínio.*

1. Introdução

A integração entre o mundo computacional e o físico tem se intensificado nas últimas décadas, especialmente com o avanço das tecnologias relacionadas aos SCF. Esses sistemas, que combinam hardware, software e redes de comunicação, têm sido aplicados em uma variedade de áreas, como veículos autônomos, automação industrial, sistemas de saúde e controle de tráfego, nas quais a necessidade de interação em tempo real com o ambiente físico é crucial (Oks *et al.*, 2024). Por definição, SCF é a união entre componentes computacionais (software), componentes físicos (hardware) e meios de comunicação, que interagem em tempo real com o ambiente, de maneira que possam monitorar, controlar ou adaptar-se ao mundo físico (Lee, 2008).

Com o crescimento exponencial de sistemas cada vez mais complexos e dinâmicos, surge a necessidade de garantir a flexibilidade, personalização e eficiência dos processos, o que é facilitado por abordagens baseadas em LPS. Uma LPS é um paradigma de desenvolvimento que visa reutilizar sistematicamente ativos de software para criar uma família de sistemas semelhantes/comuns, mas com variações específicas (Pohl *et al.*, 2005).

O paradigma de LPS surge como uma solução estratégica que permite a reutilização de ativos de software em famílias de produtos, adaptando os componentes para necessidades específicas sem perder a eficiência ou qualidade. Contudo, em SCF, a integração de componentes de hardware e software impõe desafios adicionais, como a gestão da variabilidade dos sistemas. Esse tipo de gestão deve ser capaz de acomodar diferentes versões do sistema, variando de acordo com as configurações de sensores, atuadores e requisitos específicos de cada aplicação (Fadhlillah, 2022).

Esta pesquisa investiga a intersecção entre LPS e SCF. O Gerenciamento de Variabilidade (GV) é um recurso crucial para lidar com a diversidade e os desafios dos sistemas de software modernos ou contemporâneos que estão em constante evolução. Para tanto, foi realizado um mapeamento sistemático da literatura sobre o tema, com o objetivo de compreender como as LPS têm sido aplicadas, adaptadas, modeladas e gerenciadas em ambientes ciberfísicos, além de explorar as abordagens, frameworks e plataformas mais utilizadas para facilitar essa integração.

Diante da complexidade e da crescente aplicação das LPS em SCF, tornou-se necessário compreender, de forma sistemática e abrangente, como essas duas áreas têm sido abordadas na literatura científica. Para isso, optou-se por utilizar o mapeamento sistemático como método de pesquisa, uma vez que ele permite identificar, categorizar e analisar tendências, lacunas e contribuições relevantes em um determinado campo de estudo.

O mapeamento sistemático segue um método de pesquisa estruturado com base no protocolo de Petersen (2015). Uma das primeiras premissas do protocolo é definir três questões de pesquisa principais (QPs) para conduzir o mapeamento sistemático:

- **QP1:** Como as Linhas de Produtos de Software (LPS) estão sendo aplicadas ou adaptadas no contexto de Sistemas Ciberfísicos (SCF)?
- **QP2:** Como é realizada a Gestão de Variabilidade (GV) de LPS em Sistemas Ciberfísicos?
- **QP3:** Frameworks e plataformas que utilizam LPS em SCF, e como esses frameworks tratam a variabilidade e a integração em SCF.

A condução do mapeamento envolveu uma análise quantitativa detalhada dos estudos primários encontrados em bases de dados científicas, buscando identificar soluções de GV aplicadas a sistemas e a integração de diferentes componentes de hardware e software.

Os principais resultados e discussões do mapeamento são: as abordagens encontradas, soluções desenvolvidas, tecnologias utilizadas e as maiores dificuldades encontradas durante os estudos, bem como os cenários que se aplicam para identificar as melhores práticas no processo de personalização e adaptação de SCF. A discussão dos principais

resultados revelaram tanto as vantagens quanto os desafios da adoção de LPS em SCF, oferecendo uma compreensão quantitativa das estratégias que podem ser adotadas para otimizar a produção e a evolução de SCF.

Ao longo deste trabalho as seções estão organizadas da seguinte forma: na seção 2 é apresentada a fundamentação teórica deste trabalho; na seção 3 é descrita a metodologia utilizada durante o mapeamento sistemático; na seção 4 são apresentados os principais resultados encontrados; e na seção 5, é realizada uma breve discussão sobre os resultados. Por fim, na seção 6, as conclusões são apresentadas.

2. Background

2.1. Sistemas Ciberfísicos (SCF)

Os SCF são sistemas que integram componentes computacionais, físicos e de rede para monitorar e controlar a parte física (hardware) em tempo real (Lee, 2008).

Esses sistemas operam com interações dinâmicas entre os componentes de software e os processos físicos com sua utilização em crescimento em diversas áreas, incluindo automação industrial, saúde, energia, veículos autônomos, e cidades inteligentes (Lee, 2008):

- **Veículos autônomos:** Integrando sensores, atuadores e algoritmos de decisão, os veículos autônomos são capazes de interagir com o ambiente e tomar decisões sem intervenção humana, com sistemas complexos de controle de movimento e navegação.
- **Sistemas industriais:** A automação de processos industriais é exemplo típico de SCF que monitoram, controlam e otimizam processos industriais, coletando e analisando dados em tempo real.
- **Dispositivos de saúde:** Como marcapassos ou dispositivos de monitoramento de pacientes, que podem alterar suas operações com base em condições físicas ou ambientais em tempo real.

A Figura 1 ilustra as principais relações entre um domínio específico das aplicações de SCF que é Sistema de Produção Cibe-Físicos (SPCF).



Figura 1. Capacidades de um SCF e SPCF.
Traduzido de Cardin (2019).

A Computação envolve a união da camada física com os elementos de computação e controle, enquanto a camada cibernética conecta comunicação e computação. Já o Controle possibilita a integração dos sistemas de informação por meio da relação entre controle e comunicação. Por fim, a Comunicação estabelece o elo entre os sistemas de informação e os recursos de computação da camada cibernética (Cardin, 2019).

2.2. Diferença entre Internet das Coisas (IoT) e SCF

De acordo com *Lesch et al. (2023)*, a principal diferença entre IoT e SCF está no foco e na forma de integração entre o físico e o digital. A Internet das Coisas (IoT) é centrada na identificação, comunicação e interação entre entidades físicas que passam a se conectar em rede, muitas vezes para monitoramento e coleta de dados do ambiente, podendo agir com base em gatilhos externos ou lógicas locais simples. Já os SCF são caracterizados pela integração estreita entre componentes físicos e cibernéticos, nos quais os elementos computacionais realizam processamento complexo, automação e controle em tempo real. Assim, enquanto a IoT enfatiza a conectividade e a interação com o ambiente, os SCF se distinguem por incorporar loops de realimentação e controle, em que os componentes físicos afetam os cibernéticos e vice-versa, garantindo a coordenação contínua entre os dois domínios.

2.3. Linhas de Produto de Software (LPS)

Segundo Clements e Northrop (2001) as LPS é um paradigma que visa o desenvolvimento de uma família de produtos de software a partir de um conjunto comum de artefatos reutilizáveis. Em vez de criar um produto individualmente, o objetivo das LPS é aplicar a reutilização de componentes comuns, enquanto permite a personalização e a adaptação. Há vários elementos-chave próximos a LPS:

- **Variabilidade:** É a capacidade de um sistema ser modificado ou adaptado para criar diferentes variantes ou configurações. Em LPS, a variabilidade é gerenciada de forma explícita, o que permite a criação de diversas versões do sistema a partir de uma base comum de código ou ativos (*core assets*).
- **Modelos de Características** (do inglês, *Feature Model*): São representações abstratas das variações em um sistema, permitindo que diferentes variantes de um produto sejam geradas a partir de um modelo base.
- **Arquitetura Modular:** A arquitetura de LPS é projetada para ser modular, com componentes que podem ser incluídos ou excluídos com base nos requisitos específicos de cada variante do produto.

2.4. Gestão de Variabilidade (GV)

A Gestão de Variabilidade (GV) se refere ao conjunto de práticas utilizadas para gerenciar e controlar as diferentes variações e personalizações dentro de uma LPS. O objetivo principal da GV é permitir a criação e manutenção de diferentes versões ou configurações de um sistema a partir de um conjunto comum de artefatos, promovendo reutilização, adaptabilidade e escalabilidade (Pohl, 2005).

Dentre eles os aspectos-chave da GV são:

- **Modelos de Características (*Feature Model*):** Representam graficamente as variações possíveis de um sistema, organizando funcionalidades obrigatórias, opcionais, alternativas e dependentes. São fundamentais para visualizar e planejar diferentes configurações.
- **Resolução de Variabilidade:** Processo de seleção e combinação de características específicas com base em requisitos funcionais e não funcionais, possibilitando a geração de instâncias específicas de um sistema.
- **Ferramentas de Automação:** Soluções como o FeatureIDE auxiliam na modelagem, verificação e implementação de variabilidade, tornando o processo mais eficiente e reduzindo a propensão a erros manuais.

2.5. Intersecção de LPS e SCF

No contexto de SCF, a adoção de LPS permite o desenvolvimento de sistemas complexos de forma eficiente por meio de um conjunto reutilizável de componentes para diferentes tipos de SCF. Em sistemas industriais com múltiplos sensores e atuadores, a LPS permite criar modelos de features que capturam variações (ex.: tipos de sensores, protocolos de comunicação, requisitos de tempo real) (Iglesias *et al.* 2017).

Porém, a adoção de LPS em SCF apresenta novos desafios por causa da natureza dinâmica e interconectada dos SCF, que exigem uma integração cuidadosa entre os componentes de hardware e software (Kruger *et al.* 2017).

Os desafios no desenvolvimento de SCF incluem a necessidade de garantir baixa latência, alta confiabilidade, segurança adequada e escalabilidade viável. Por causa da natureza dinâmica e distribuída desses sistemas, o GV em SCF é ainda mais desafiador por causa da complexidade dos SCF, que envolvem tanto software quanto componentes físicos (Kruger *et al.* 2017).

Entre esses desafios encontra-se (Kruger *et al.* 2017).:

- **Integração com componentes físicos:** A variabilidade não se limita ao software; ela envolve a escolha de sensores, atuadores e outros componentes físicos, o que torna a gestão de variabilidade complexa.
- **Adaptação em tempo real:** A variabilidade precisa ser gerenciada de maneira que o SCF possa se adaptar dinamicamente ao ambiente em tempo real, sem comprometer a performance ou a segurança do sistema.
- **Garantia de segurança e confiabilidade:** Em sistemas críticos, como veículos autônomos ou sistemas médicos, a variabilidade não pode introduzir falhas ou vulnerabilidades no sistema.

Dentro desse contexto também existe a diferença entre features e variabilidade, por mais que estejam próximas as features descrevem o que pode existir no sistema enquanto as variabilidades descrevem como essas features podem mudar, ser configuradas ou adaptadas em diferentes instâncias de um SCF.

No domínio automotivo, exemplos de features incluem sensoramento do ambiente, piloto automático e tipos de motorização (combustão, elétrico ou híbrido). Essas funcionalidades apresentam variabilidades como a escolha de sensores (câmera, LIDAR, radar), os modos de condução (manual, assistido ou autônomo) e os protocolos de comunicação (4G, 5G) (Ramesh *et al.* 2017).

3. Metodologia

O método de pesquisa adotado neste trabalho é o mapeamento sistemático proposto por Petersen (2015), com o objetivo de investigar a aplicação de LPS em SCF, o processo de GV em LPS aplicados a SCF e as abordagens, frameworks e plataformas que utilizam LPS nesses sistemas.

O processo de mapeamento sistemático foi dividido em quatro etapas principais: definição das questões de pesquisa, busca e seleção dos estudos, extração de dados e análise dos dados. A Figura 2 retrata todo o processo realizado. As próximas seções detalham as etapas de tal método adotado para o contexto deste trabalho.

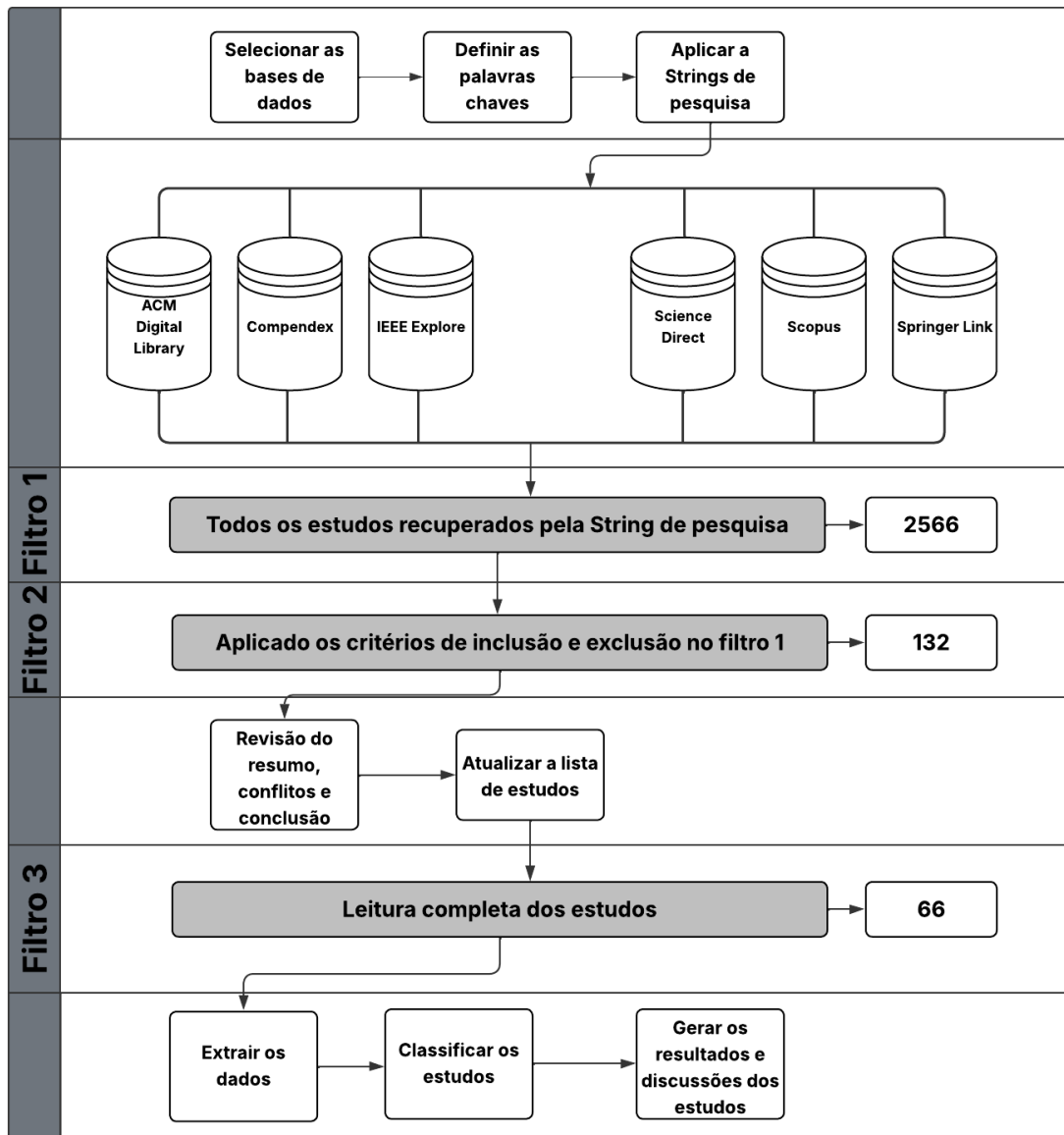


Figura 2. Fases do método de pesquisa adaptado de Software product line applied to the internet of things: A systematic literature review (Geraldi, 2020) .

3.1 Definição das Questões de Pesquisa

Este estudo aborda três Questões de Pesquisa principais (QPs) de maneira que essas Questões guiam o processo de filtragem e validação dos artigos estudados para a condução desta pesquisa, as quais são:

- **QP1:** Como as Linhas de Produtos de Software (LPS) estão sendo aplicadas ou adaptadas no contexto de Sistemas Ciberfísicos (SCF)?
- **QP2:** Como é realizada a Gestão de Variabilidade (GV) de LPS em Sistemas Ciberfísicos?
- **QP3:** Frameworks e plataformas que utilizam LPS em SCF, e como esses frameworks tratam a variabilidade e a integração em SCF.

Essas questões foram formuladas para guiar a revisão da literatura e garantir que todas as facetas importantes do uso de LPS em SCF fossem abordadas de maneira clara e estruturada.

3.2 Busca e Seleção dos Estudos

A busca e seleção dos estudos segue as diretrizes de Petersen *et al.* (2015) para garantir tal atividade como abrangente e sistemática. As etapas incluem:

- **Identificação de fontes relevantes:** A busca foi realizada em bases de dados acadêmicas e repositórios de pesquisa, como IEEE Xplore, ACM Digital Library, SpringerLink, Scopus, entre outros, utilizando palavras-chave como "Software Product Lines", "Cyber-Physical Systems", "Variability Management", "Feature Models" e outros termos relacionados.
- **Critérios de inclusão:** Estudos que abordam a aplicação de LPS em SCF, a gestão de variabilidade em SCF, e as plataformas ou frameworks para LPS em SCF foram incluídos na revisão. Os estudos deveriam ser publicados entre 2000 e 2024, e poderiam ser artigos de conferências, periódicos ou capítulos de livros.
- **Critérios de exclusão:** Trabalhos que não envolvem LPS ou SCF; que tratavam de áreas não relacionadas como modelagem de sistemas puramente teóricos ou sem aplicação prática; e trabalhos em idiomas diferentes do inglês ou português foram descartados.

3.3 Extração de Dados

A extração de dados segue uma abordagem em que as informações foram coletadas de cada estudo incluído, abrangendo os seguintes aspectos principais:

- **Aplicação de LPS em SCF (QP1):** Tipos de sistemas SCF que adotam LPS, as áreas de aplicação (automação, veículos autônomos, sistemas industriais, dentre outras) e como as LPS são aplicadas ou adaptadas a esses sistemas.
- **Gestão de Variabilidade em SCF (QP2):** Métodos e ferramentas de gestão de variabilidades aplicadas à SCF, incluindo a utilização de modelos de características, integração de variabilidade nos sistemas e os desafios enfrentados.
- **frameworks ou plataformas utilizadas (QP3):** Frameworks e plataformas que utilizam LPS em SCF, e como esses frameworks tratam a variabilidade e a integração em SCF.

3.4 Análise dos Dados

A análise dos dados considera o modelo de análise de dados proposto por Petersen *et al.* (2015), que pode ser dividido em duas etapas principais:

- **Análise quantitativa:** A análise quantitativa focou na frequência de menções a determinadas abordagens, técnicas e ferramentas nos estudos revisados. A contagem das ocorrências de palavras-chave (*keywords*), como "variabilidade", "frameworks", "testes", "complexidade", entre outras, foi

realizada para identificar as abordagens mais recorrentes e os pontos de foco nos estudos.

- **Análise qualitativa:** Para entender a aplicação e os desafios da adoção de LPS em SCF, foram analisados aspectos qualitativos como os modelos de variabilidade utilizados, a adaptação de LPS a diferentes domínios e o impacto da gestão de variabilidade nas soluções desenvolvidas.

É importante destacar que esta pesquisa de mapeamento sistemático não se aprofundou em uma discussão qualitativa na maioria dos estudos, visto que o mapeamento possui, em sua natureza, um caráter de análise quantitativo.

4. Resultados

Com base nas etapas do método de mapeamento sistemático de Petersen *et al.* (2015), QPs definidas na seção 3.1 foram utilizadas para orientar o processo de seleção. Assim, a partir dos principais resultados identificados neste mapeamento sistemático. A análise dos dados demográficos é apresentada na Figura 3, com a distribuição e fontes de informação utilizadas na pesquisa. Os dados foram extraídos de estudos primários selecionados com o objetivo de fornecer uma visão abrangente sobre a frequência, origem e tendências das publicações relevantes para este estudo.

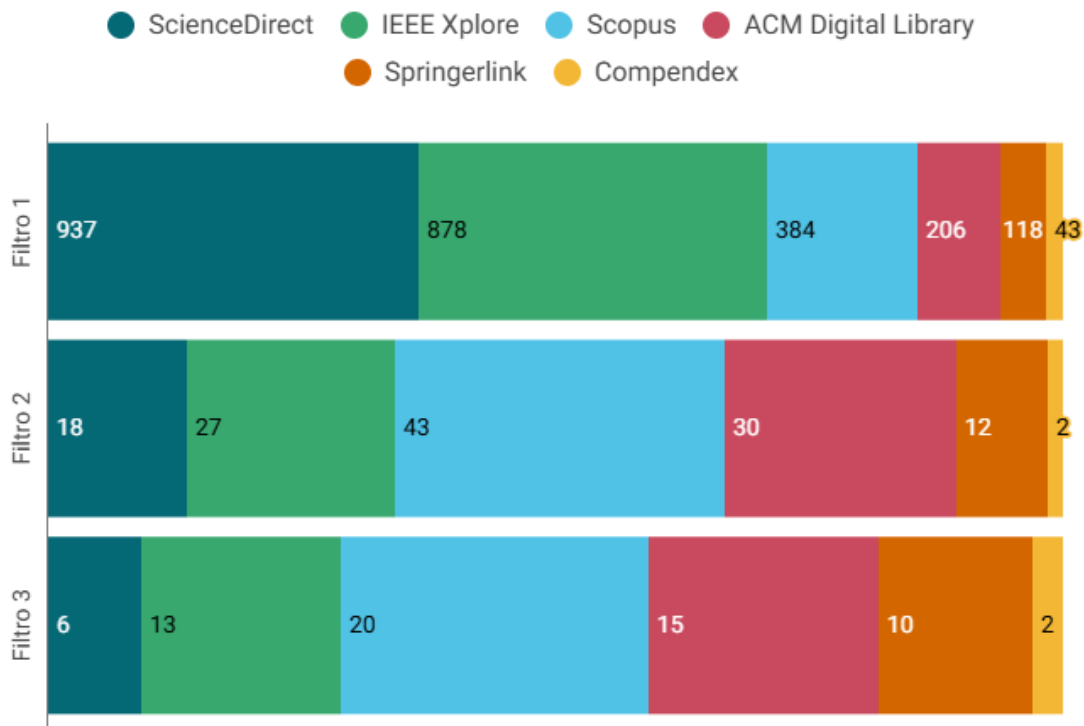


Figura 3. Quantidade de estudos recuperados com base em cada etapa da filtragem.

A Figura 3 ilustra a quantidade de artigos encontrados nas diferentes bases de dados consultadas, após a aplicação sucessiva de três filtros de seleção. Observa-se que, na primeira etapa (Filtro 1), as bases contam com o número de artigos recuperados igual a ScienceDirect (937 estudos), IEEE Xplore (878 estudos) e Scopus (384 estudos), ACM Digital Library (206 estudos), Springerlink (118 estudos) e Compendex (43 estudos). Esses números representam todos os estudos recuperados pela string de pesquisa como mostrado na figura 2.

No Filtro 2, houve uma redução significativa após a aplicação dos critérios de inclusão e exclusão no filtro 1, ScienceDirect (18 estudos), IEEE Xplore (27 estudos) e Scopus (43 estudos), ACM Digital Library (30 estudos), Springerlink (12 estudos) e Compendex (2 estudos).

No Filtro 3, a quantidade de artigos foi reduzida após a revisão do resumo, conflitos e conclusão dos estudos do filtro 2, ScienceDirect (6 estudos), IEEE Xplore (13 estudos) e Scopus (20 estudos), ACM Digital Library (15 estudos), Springerlink (10 estudos) e Compendex (2 estudos).

A Figura 4 evidencia o processo de refinamento dos resultados por meio dos filtros, demonstrando a eficácia dos critérios de inclusão e exclusão aplicados. Além disso, permite visualizar de forma clara quais bases de dados contribuíram com maior relevância para a amostra final de estudos utilizados na pesquisa.

Como resultado, identificamos e selecionamos 66 estudos primários (Filtro 3) distribuídos entre seis bases de dados digitais entre o período de 2000 e 2024. A Figura 4 ilustra a quantidade de estudos encontrados em cada uma dessas bases de dados.

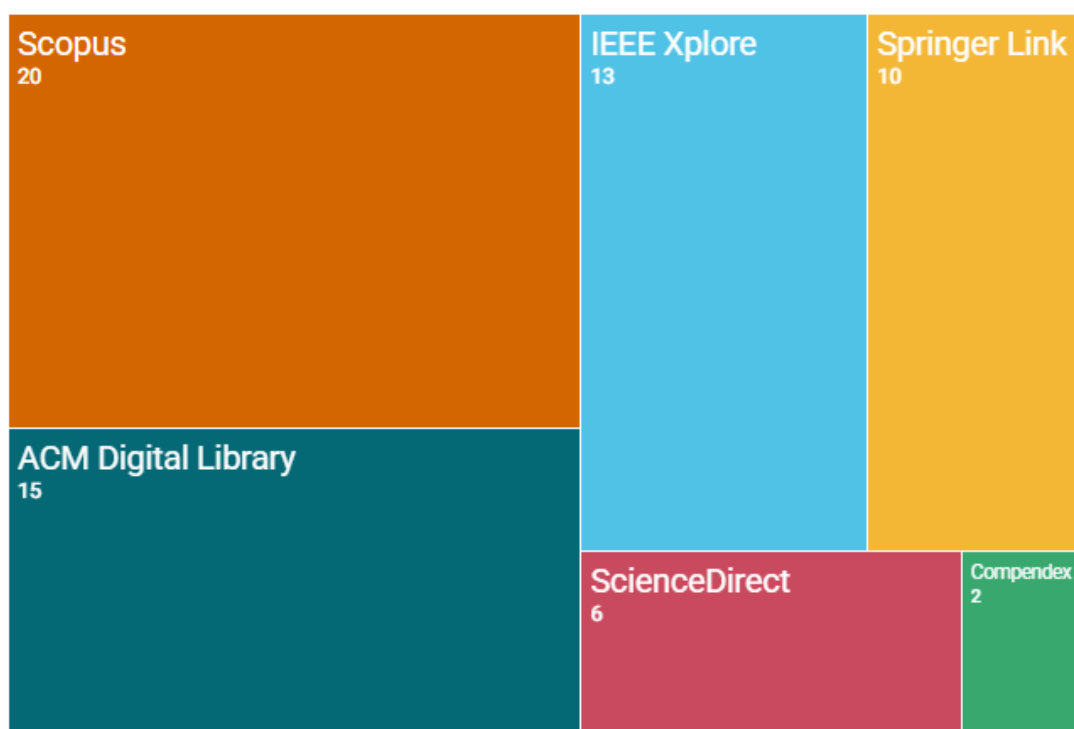


Figura 4. Quantidade de estudos encontrados por base de dados.

A Figura 5 sendo um complemento da Figura 4, apresenta a distribuição dos estudos selecionados ao longo dos anos, segmentada por base de dados distintas. O gráfico da Figura 5 fornece uma visão detalhada da evolução das publicações acadêmicas relacionadas ao tema analisado, permitindo identificar tendências e padrões na produção científica ao longo do tempo.

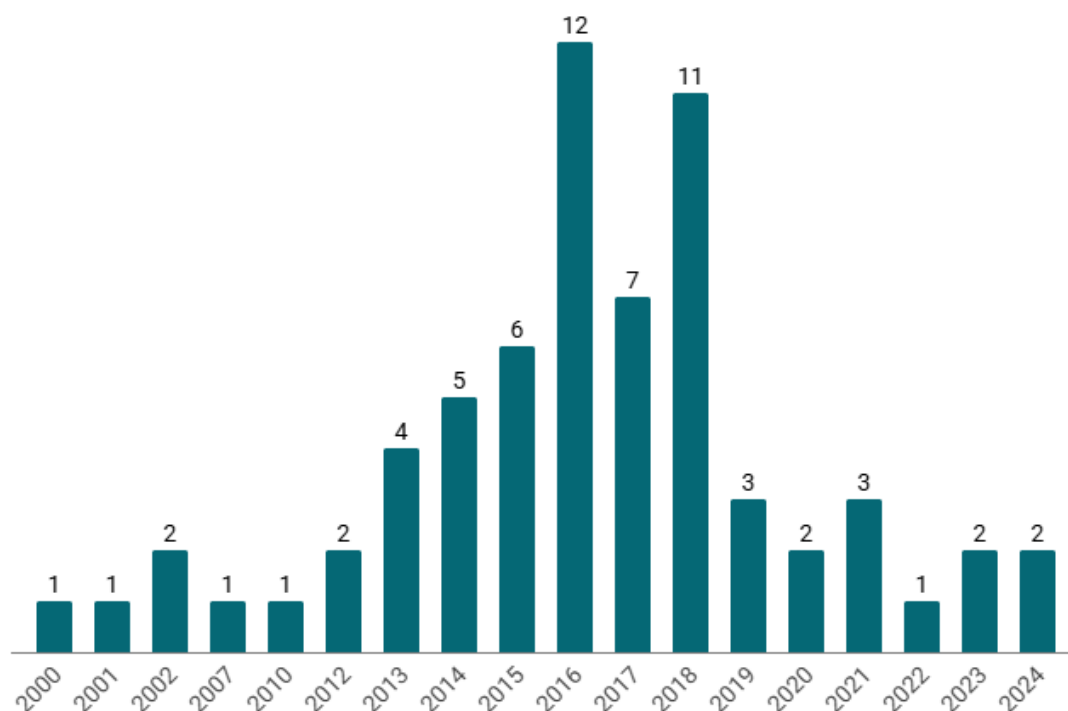


Figura 5. Análise de Estudos por Ano.

Observa-se na Figura 5 que a quantidade de estudos varia significativamente entre os anos. Segundo o estudo realizado nas últimas 2 décadas, há uma baixa frequência de publicações indexadas a partir de 2000, o que pode ser atribuído a um menor volume de pesquisas formais sobre o tema porque até 2008 não há uma definição concreta para o termo SCF.

A partir de 2012, temos indícios de crescimento nas amostragens, evidenciando um aumento nas pesquisas na área, possivelmente sendo uma consequência direta da evolução da tecnologia em que começou a ser mais impactante a necessidade de lidar com os problemas de escalonamento dos SCF como a dificuldade em garantir desempenho em tempo real, a coordenação entre múltiplos sensores e atuadores distribuídos, e a alocação eficiente de recursos computacionais. Isso pode ter ocasionado o desenvolvimento de mais pesquisas nessa área. Entretanto, a partir de 2019 nota-se uma distribuição mais dispersa dos estudos, com publicações ainda presentes, embora em menor número. Essa redução pode indicar uma desaceleração momentânea no volume de pesquisas, talvez a dificuldade da complexidade na

modelagem de SCF e por ser multidisciplinar, mas não permite afirmar com certeza uma tendência de queda.

Observando os principais estudos primários, foram utilizadas estratégias diferentes para lidar com os problemas de controle de variabilidade como mostrado na Figura 6. As respostas desses problemas se resumem ao desenvolvimento de modelos resultantes que possam categorizar as variabilidades e possibilitar a sua modularização de maneira que variantes semelhantes sejam trabalhadas dentro do mesmo escopo.

Frameworks	Ferramentas	Abordagens	Técnicas
SimPL	FeatureIDE	DSPL (Dynamic Software Product Lines)	CVL (Common Variability Language)
Zen-DO / Zen-Configurator	ASTERYSCO	MDSE (Model-Driven SE)	FM (Feature Modeling)
GRAF	SPOT + Cloud Computing	MBD (Model Based Design)	Delta Simulink
FCORE	MATLAB/Simulink	PLE (Product Line Engineering)	Kumbang
AUTOSAR			MVVM (Multi-View Variability Model)
QUASAR			CIT (Combinatorial Interaction Testing)
EMF (Eclipse Modeling Framework)			Delta-Oriented Modeling

Figura 6. Representação de Todas as Abordagens, Plataformas e Técnicas encontradas.

Analisando a Figura 6, tecnologias em comuns tiveram as mesmas perspectivas parecidas de soluções para os seus problemas de variabilidade. A tecnologia mais comum encontrada é a ferramenta Simulink. Essa ferramenta permite tanto modelar, como também realizar a simulação e análise de sistemas dinâmicos em ambientes como o MATLAB, segundo exemplo de Haber (2013).

Dentro do processo do estudo de Haber (2013) pode haver estratégias quanto à prática da modelagem, como exemplo da Delta Simulink. A Delta Simulink oferece uma interface gráfica para criação e aplicação de deltas no Simulink de maneira que os blocos variáveis são encapsulados e podem ser aplicados modularmente para alterar o comportamento do sistema. Um framework encontrado que vale ressaltar é SimPL que é voltado para modelar variabilidade e comunalidade em sistemas integrados de controle (ICS). Essa metodologia tem como ênfase o uso da UML e do Feature Modeling (FM). Em adição, o estudo de Safdar (2016) organiza as funcionalidades de um sistema em um diagrama hierárquico que descreve quais funcionalidades são obrigatórias, opcionais ou alternativas e que podem coexistir em uma família de produtos.

A Figura 7 representa a incidência das abordagens/ferramentas mais recorrentes nos artigos estudados.

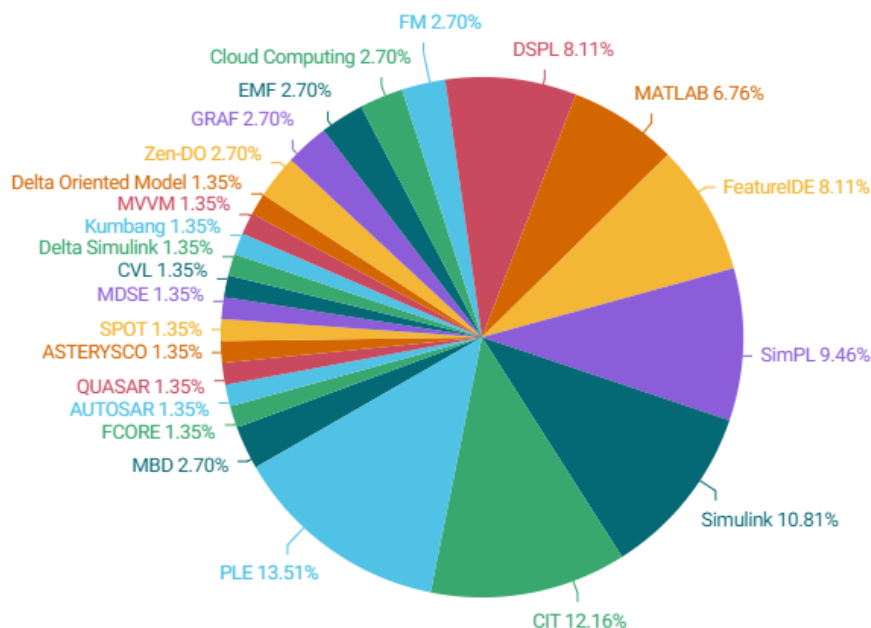


Figura 7. Incidência das abordagens mais recorrentes nos artigos estudados.

5. Discussão dos Principais resultados

Conforme o contexto do SCF, a medida que grandes LPS evoluem, leva a uma degradação do seu modelo resultante exigindo refatorações para se adequar novamente. Desafios mais claros encontrados: gerenciamento de variabilidade em múltiplas camadas, validação/testes (combinatorial testing, delta testing), adaptação dinâmica (DSPL).

Pode haver também dificuldade em gerenciar um alto número de variabilidade devido a falta de ferramentas adequadas, ou de gerenciar a evolução consistente de alguns softwares à medida que novas funcionalidades vão surgindo especialmente com a necessidade de integrar novos módulos sem comprometer a performance e segurança do sistema já existente. Há também o fato de que as novas variantes que vão surgindo podem deixar mais complexo o processo de testar todas elas.

A análise dos principais desafios em LPS para SCF revela que a complexidade e a variabilidade são os problemas mais recorrentes. A dificuldade em gerenciar variantes, configurar corretamente os sistemas e realizar testes eficazes impacta diretamente o tempo e a confiabilidade das soluções. Em adição, a necessidade de otimização e o uso de simulação aparecem como fatores relevantes, embora menos frequentes. Esses

desafios reforçam a importância da automação, modelagem avançada e estratégias eficientes de teste para garantir a qualidade e a escalabilidade das LPS em SCF.

A seguir, a Figura 8 apresenta a frequência das palavras-chave mais mencionadas nos estudos realizados.

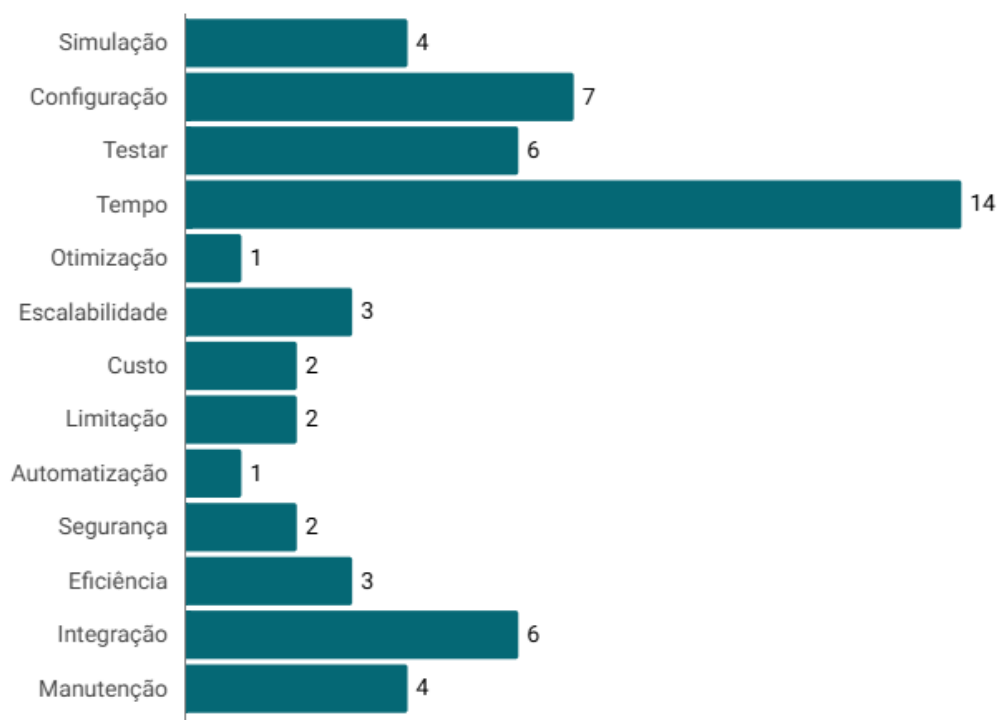


Figura 8. Palavras-Chave mais recorrentes nos desafios encontrados.

Observando a Figura 8, a complexidade aparece com 26 ocorrências como o tema mais recorrente, sendo resultado da natureza multifacetada dos SCF, combinada com as estratégias de reutilização e adaptação típicas das LPS que geram altos níveis de complexidade. A complexidade se dá por conta da dificuldade de gerenciar todo o desenvolvimento das variantes que precisam atender múltiplos domínios (mecânico, elétrico e computacional) e sua integração com uma diversidade de requisitos entre produtos.

Com 15 ocorrências, o tempo se relaciona com todo tempo gasto relacionado ao GV, desde tempo de resposta (software), tempo real (hardware) e todo tempo de desenvolvimento ou configuração. Isso ocorre por conta do desenvolvimento ser muito custoso já que se deve simular, configurar e testar todos os requisitos necessários.

Durante a análise e interpretação dos estudos primários selecionados neste mapeamento sistemático, foram identificados diversos problemas no processo de criar uma LPS em SCF. O GV está presente em várias etapas do processo, dentre eles o resultado final, manutenção dos modelos resultantes e os testes realizados, como podem ser observados nas Figuras 9, 10 e 11.

Em relação à manutenção dos modelos resultantes os principais problemas, destacam-se:

- **Número de Variantes Aumenta:** Conforme a LPS cresce, novas funcionalidades e personalizações são adicionadas para atender diferentes requisitos e cenários operacionais. Esse crescimento resulta em um número exponencial de variantes, tornando difícil rastrear, organizar e validar todas as possibilidades (Arrieta *et al.* 2018).
 - **Gerenciar um Enorme Número de Variantes:** Com a multiplicidade de variantes, a gestão eficaz se torna um grande desafio. A complexidade aumenta devido à necessidade de manter compatibilidade entre diferentes configurações, garantir que todas as variações sejam corretamente documentadas e que possam ser testadas e verificadas de maneira eficiente (Arrieta *et al.* 2018).
 - **Complexidade de Manutenção de Grandes LPS a Medida Que Evoluem Levando à Degeneração do Modelo Resultante:** À medida que a LPS evolui, sem um gerenciamento adequado, o modelo resultante pode se tornar inconsistente e difícil de manter. Isso leva à degeneração do modelo resultante, onde dependências ocultas, código duplicado e configurações conflitantes começam a emergir, comprometendo a escalabilidade e a qualidade do sistema (Metzger *et al.*, 2016).
- **Dificuldade na Separação entre Variabilidade e Funcionalidade em Modelos Resultantes de Grande Porte:** Em modelos resultantes de grande porte, a linha entre variabilidade e funcionalidade principal pode se tornar difusa. Isso ocorre porque a variabilidade pode ser distribuída em diferentes partes do sistema, dificultando a compreensão do que é essencial e o que é opcional ou específico para determinadas configurações (Iglesias *et al.* 2017)

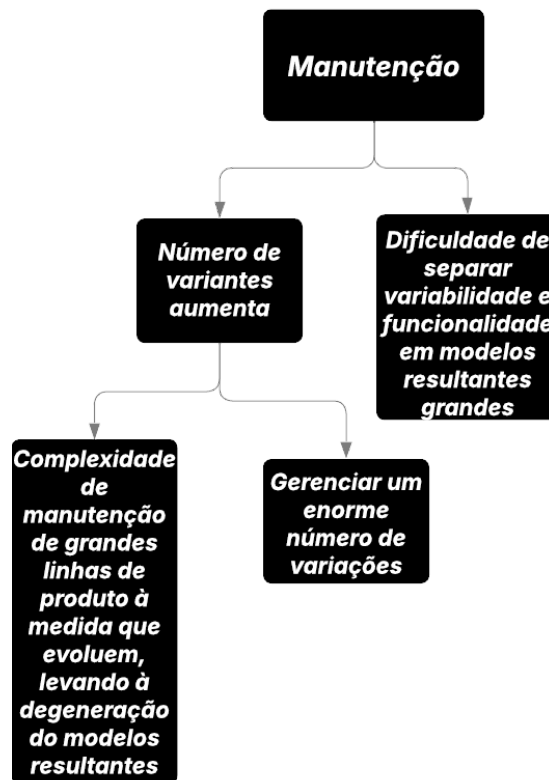


Figura 9. Problemas na fase de Gerenciamento da Manutenção.

O mesmo ocorre para o processo de testes durante a criação do modelo resultante, um barreira que apresenta desafios significativos devido à complexidade e à variabilidade inerente a esses sistemas. Entre os principais problemas enfrentados, destacam-se:

- **Complexidade de Testar Manualmente Todas as Variantes:** A grande quantidade de configurações possíveis em um SCF configurável torna inviável a realização de testes manuais exaustivos. Cada variante pode ter comportamentos distintos, exigindo um esforço elevado para garantir que todas as combinações sejam validadas adequadamente (Arrieta *et al.* 2018).
- **Dificuldade em Testar Todas as Variantes:** Mesmo com a automação de testes, garantir a cobertura total de todas as variantes é um desafio. A quantidade de combinações das opções configuráveis leva a um número exponencial de cenários a serem testados, dificultando a identificação de quais variantes devem ser priorizadas para garantir a confiabilidade do sistema sem comprometer o tempo e os recursos disponíveis (Markiegi *et al.* 2017).
- **Falta de Confiabilidade do Escopo:** Definir corretamente o escopo dos testes dentro de um SCF altamente variável é um problema crítico. Com a evolução contínua da LPS, é difícil garantir que todas as interações entre variantes sejam testadas de forma adequada, levando ao risco de falhas não detectadas em determinadas combinações (Cortiñas *et al.* 2017).

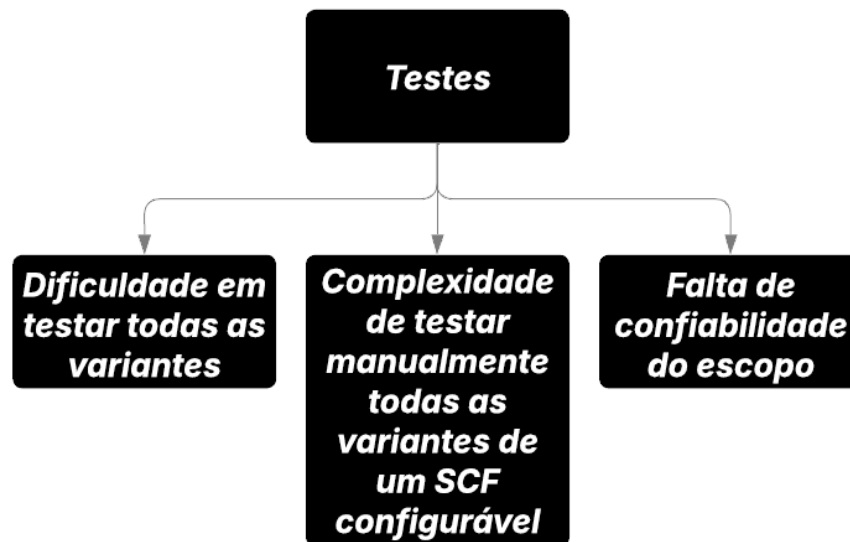


Figura 10. Problemas na Fase de Testes.

Por fim, avaliar a confiabilidade dos modelos resultantes gerados pela aplicação de LPS em SCF tem sido um desafio, principalmente pela dificuldade em desenvolver modelos resultantes concretos e generalizáveis para diferentes setores industriais. Entre os principais obstáculos estão:

- **Dificuldade de Generalizar uma Solução de LPS para Múltiplos Domínios Industriais:** Cada setor industrial possui requisitos específicos, padrões regulatórios e particularidades operacionais que dificultam a aplicação de uma única abordagem LPS. A adaptação da LPS para diferentes domínios exige um esforço significativo para garantir compatibilidade, flexibilidade e aderência às necessidades de cada indústria (Goiuria, 2017).
- **Configurações Corretas:** A escolha de configurações adequadas dentro de um SCF configurável é um desafio crítico. A falta de diretrizes claras para determinar quais configurações são mais eficientes e seguras pode levar a problemas de desempenho, falhas inesperadas ou incompatibilidades em determinados ambientes operacionais (Arrieta *et al.* 2014).
- **Falta de Resultados Concretos:** Apesar do avanço das pesquisas em LPS para SCF, ainda há uma carência de estudos empíricos e implementações práticas que comprovem a eficácia dessas abordagens em larga escala. A

ausência de benchmarks padronizados e estudos de caso amplamente validados dificulta a adoção da LPS em contextos industriais reais (Arrieta *et al.* 2014).

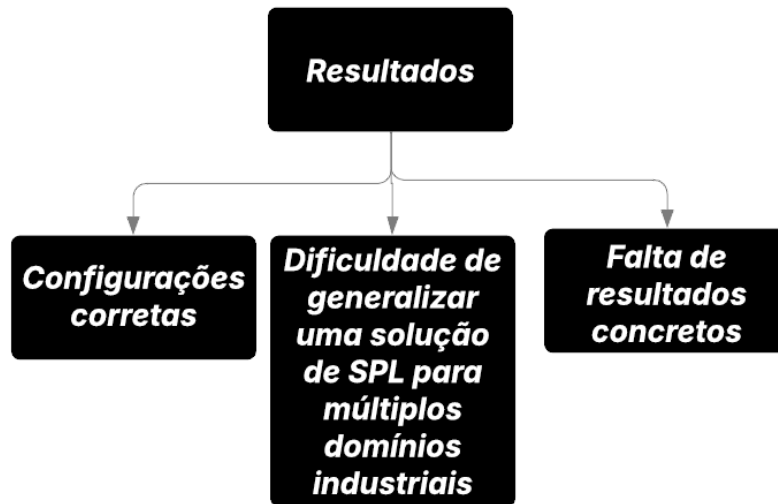


Figura 11. Principais Resultados encontrados.

Embora as LPS e o GV oferecem grandes benefícios em termos de reutilização e customização, sua implementação em SCF apresenta os principais desafios significativos:

- **Escalabilidade e complexidade:** SCF envolvem múltiplos componentes distribuídos e heterogêneos, o que torna difícil gerenciar a variabilidade em larga escala (Lauber *et al.*, 2018).
- **Gerenciamento de requisitos dinâmicos:** Os SCF frequentemente operam em ambientes dinâmicos onde os requisitos podem mudar ao longo do tempo, exigindo uma gestão de variabilidade flexível e adaptativa (Safdar *et al.*, 2016).
- **Interoperabilidade:** Integrar componentes de diferentes fornecedores ou tecnologias (como sensores, atuadores e software) em um sistema coeso é um desafio complexo em SCF (Beek *et al.*, 2018).
- **Custo e tempo de desenvolvimento:** A adaptação e personalização de um SCF utilizando LPS podem ser caras e demoradas, especialmente quando se trabalha com requisitos de tempo real e integração com componentes físicos (Arrieta *et al.* 2017).

A integração entre Linhas de Produtos de Software (LPS) e Sistemas Ciberfísicos (SCF) tem sido tratada na literatura principalmente pela evolução das tradicionais SPLs para Dynamic Software Product Lines (DSPLs). Essa abordagem amplia o escopo das LPS ao permitir configuração e reconfiguração em tempo de execução, característica essencial em SCF, uma vez que estes operam em ambientes dinâmicos e imprevisíveis. Trabalhos como o de SmartyCo mostram como essa estratégia é aplicada em casas inteligentes, permitindo a reconfiguração de sensores e atuadores de acordo com o contexto de uso (Romero *et al.* 2015).

Além disso, os estudos destacam a importância de modelos de variabilidade dinâmicos, que vão além da resolução de inconsistências apenas em tempo de modelagem, como ocorre em ferramentas estáticas (por exemplo, o FeatureIDE). Em SCF, é necessário que os modelos de características sejam capazes de refletir continuamente o estado atual do sistema e permitir a avaliação incremental das restrições. Nesse sentido, linguagens como a Object Constraint Language (OCL) têm sido utilizadas para verificar consistência em configurações durante a execução, como nos trabalhos que apresentam o Zen-Configurator (Yue *et al.* 2016).

A literatura também evidencia que, para SCF, não basta apenas reconfigurar: é necessário validar em tempo real. Assim, estratégias de testes contínuos têm sido aplicadas, incluindo o uso de Combinatorial Interaction Testing (CIT) e algoritmos de busca para priorização de casos de teste, garantindo que a cobertura seja mantida mesmo diante da multiplicidade de variantes (Markiegi *et al.* 2017). Ferramentas como o ASTERYSCO avançam nesse ponto ao permitir a geração automática de instâncias de teste em diferentes configurações de SCF (Arrieta *et al.* 2017).

Por fim, observa-se a necessidade de integração com frameworks de domínio. Diferentemente de LPS puramente de software, SCF exigem compatibilidade com padrões consolidados, como o AUTOSAR na indústria automotiva e o MATLAB/Simulink no desenvolvimento e simulação de modelos incrementais. Trabalho como o de Thiel *et al.* (2002) destaca justamente essa interseção entre metodologias de LPS e frameworks específicos de SCF.

6. Conclusão

Esta pesquisa investigou a interseção entre LPS e SCF, por meio de um mapeamento sistemático da literatura baseado nas maneiras que vêm sendo aplicadas ou adaptadas LPS em SCF, como realizada sua GV e quais abordagens, frameworks e plataformas utilizam LPS em SCF. A análise dos estudos selecionados revelou que, embora as LPS ofereçam benefícios substanciais em termos de reutilização de componentes, personalização e redução de custos, a sua aplicação em SCF apresenta desafios significativos devido à complexidade de SCF e à necessidade de integração entre componentes de hardware e software.

Ferramentas como o FeatureIDE (2014) já demonstravam que a introdução de constraints em modelos de características é suficiente para resolver inconsistências na fase de modelagem, eliminando configurações inválidas antecipadamente. Contudo, no

contexto de Sistemas Ciberfísicos, persiste o desafio de estender esse controle para adaptações em tempo de execução, onde a variabilidade é dinâmica.

Os principais resultados apontam que o GV em LPS para SCF deve ser cuidadosamente gerida porque abrange não apenas a configuração e personalização de software, mas a adaptação dinâmica de componentes físicos em tempo real. As soluções mais comuns para lidar com esses desafios envolvem a utilização de modelos de características, como FeatureIDE e SimPL, que permitem uma abordagem modular e flexível, capaz de acomodar as diferentes variações entre os sistemas. No entanto, questões como a escalabilidade, a interoperabilidade entre diferentes tecnologias e a segurança permanecem obstáculos que precisam ser superados para a adoção mais ampla dessas abordagens.

A integração das metodologias de Engenharia de Sistemas Baseada em Modelos (MBSE), Product Line Engineering (PLE) e ferramentas de modelagem, como Simulink, demonstraram grande potencial para ajudar no controle de variabilidade e na redução da complexidade em SCF de grande escala, como veículos autônomos e sistemas industriais. Contudo, os custos de simulação e a dificuldade de testar todas as configurações possíveis de um SCF de forma eficiente ainda são desafios significativos que impactam a viabilidade e o tempo de desenvolvimento desses sistemas.

Por fim, este trabalho contribui para a compreensão de como LPS podem ser aplicadas de forma eficaz em SCF, identificando soluções já existentes, apontando as lacunas e desafios que precisam ser enfrentados em futuras pesquisas.

7 - Referências

Arrieta, A.; Sagardui, G.; Etxeberria, L. (2014). A configurable test architecture for the automatic validation of variability-intensive cyber-physical systems. (SPLC 2014)

Arrieta, A.; Sagardui, G.; Etxeberria, L. (2014). Towards the automatic generation and management of plant models for the validation of highly configurable cyber-physical systems. (SPLC 2014) Workshops.

Arrieta, A.; Wang, S.; Sagardui, G.; Etxeberria, L. (2016). Test case prioritization of configurable cyber-physical systems with weight-based search algorithms. (GECCO 2016).

Arrieta, Aitor; Sagardui, Goiuria; Etxeberria, Leire; Zander, Justyna (2017). Automatic generation of test system instances for configurable cyber-physical systems. (SPLC 2017).

Arrieta, A.; Wang, Shuai; Sagardui, Goiuria; Etxeberria, Leire (2016). Search-based Test Case Selection of Cyber-physical System Product Lines for Simulation-based Validation. (SSBSE 2016).

Aitor A., Shuai Wang, Goiuria Sagardui, Leire Etxeberria (2018). Search-Based Test Case Prioritization for Simulation-Based Testing of Cyber-Physical System Product Lines. (ICST 2018).

Asikainen, Timo; Männistö, Tomi; Soininen, Timo (2007). Kumbang: A domain ontology for modelling variability in software product families. (SPLC 2007)

Becker, Martin; Zhang, Bo (2018). How Do Our Neighbours Do Product Line Engineering?: A Comparison of Hardware and Software Product Line Engineering Approaches from an Industrial Perspective. (SPLC 2018)

Beek, Maurice H. ter; Fantechi, Alessandro; Gnesi, Stefania (2018). Product Line Models of Large Cyber-physical Systems: The Case of ERTMS/ETCS. (SPLC 2018)

Bordel, B.; Iturrioz, T.; Alcarria, R.; Sanchez-Picot, A. (2018). Provision of next-generation personalized cyber-physical services. (Future Generation Computer Systems, Elsevier)

Buchmann, Thomas; Baumgartl, Johannes; Henrich, Dominik; Westfechtel, Bernhard (2015). Robots and Their Variability: A Societal Challenge and a Potential Solution. (SPLC 2015)

Bures, Tomas; Hnetynka, Petr; Plasil, Frantisek (2014). Strengthening Architectures of Smart CPS by Modeling Them as Runtime Product-lines.

Cleland-Huang, J.; Vierhauser, M.; Bayley, S. (2018). Dronology: An incubator for cyber-physical systems research. (RE 2018 – IEEE International Requirements Engineering Conference)

Clements e Northrop (2001). Software product lines: practices and patterns. (Livro – Addison-Wesley)

Cortiñas, Alejandro; Luaces, Miguel R.; Pedreira, Oscar; Places, Ángeles S.; Pérez, Jennifer (2017). Web-based Geographic Information Systems SPLE: Domain Analysis and Experience Report. (SPLC 2017)

Derakhshanmanesh, Mahdi; Salehie, Mazeiar; Ebert, Jurgen (2012). Towards model-centric engineering of a dynamic access control product line. (SPLC 2012)

Di Cola, S.; Tran, C.; Lau, K.-K.; Qian, C.; Schulze, M. (2016). A Component Model for Defining Software Product Families with Explicit Variation Points. (SPLC 2016)

Dominka, S.; Ertl, D.; Dubner, M.; Wiesinger, R.; Kaindl, H. (2018). Taming and optimizing feature interaction in software-intensive automotive systems. (SPLC 2018)

Haber, A.; Kolassa, C.; Manhart, P.; Nazari, P.M.S.; Rumpe, B.; Schaefer, I. (2013). First-class variability modeling in Matlab/Simulink. (SPLC 2013)

Haber, Arne; Rendel, Holger; Rumpe, Bernhard; Schaefer, Ina (2012). Evolving Delta-Oriented Software Product Line Architectures. (SPLC 2012)

Hein, Andreas; Schlick, Michael; Vinga-Martins, Renato (2000). Applying Feature Models in Industrial Settings. (SPLC 2000)

- Hellebrand, R.; Schulze, M.; Becker, M. (2016). A branching model for variability-affected cyber-physical systems. (SPLC 2016)
- Iglesias, Aitziber; Lu, Hong; Arellano, Cristóbal; Yue, Tao; Ali, Shaukat; Sagardui, Goiuria (2017). Product Line Engineering of Monitoring Functionality in Industrial Cyber-Physical Systems: A Domain Analysis. (SPLC 2017)
- Käßmeyer, M.; Bazan, P.; Schurius, M.; Berndt, R.; German, R. (2016). A Formal Model for Stateful and Variant-Rich Automotive Functions. (SPLC 2016)
- Käßmeyer, M.; Berndt, R.; Bazan, P.; German, R. (2016). Product line fault tree analysis by means of multi-valued decision diagrams. (ISSRE 2016)
- Krüger, Jacob; Nielebock, Sebastian; Krieter, Sebastian; Diedrich, Christian; Leich, Thomas; Saake, Gunter; Zug, Sebastian; Ortmeier, Frank (2017). Beyond Software Product Lines: Variability Modeling in Cyber-Physical Systems. (SPLC 2017)
- Lauber, A.; Guissouma, H.; Sax, E. (2018). Virtual Test Method for Complex and Variant-Rich Automotive Systems. (SPLC 2018)
- Lee (2008). Cyber Physical Systems: Design Challenges. (ISORC 2008)
- Lesch, V., Züfle, M., Bauer, A., Iffländer, L., Krupitzer, C., & Kounev, S. (2023). A literature review of IoT and CPS—What they are, and what they are not. (Journal of Systems and Software – Elsevier, 2023)
- Li, M.; Guan, L.; Dickerson, C.; Grigg, A. (2016). Model-based systems product line engineering with physical design variability for aircraft systems. (Procedia CIRP, Elsevier, 2016)
- Lu, Hong; Yue, Tao; Ali, Shaukat; Zhang, Li (2016). Model-based incremental conformance checking to enable interactive product configuration. (MODELS 2016)
- Mansoor, Niloofar; Saddler, Jonathan A.; Silva, Bruno; Bagheri, Hamid; Cohen, Myra B.; Farritor, Shane (2018). Modeling and testing a family of surgical robots: An experience report. (SPLC 2018)
- Markiegi, Urtzi (2017). Test Optimisation for Highly-Configurable Cyber-Physical Systems. (SPLC 2017)
- McGee, Ethan T.; McGregor, John D. (2015). Composition of Proof-carrying Architectures for Cyber-physical Systems. (SPLC 2015)
- McGee, Ethan T.; Silva, Roselane S.; McGregor, John D. (2017). Composition of Verification Assets for Software Product Lines of Cyber Physical Systems. (SPLC 2017)
- Metzger, A.; Bayer, A.; Doyle, D.; Sharifloo, A. M.; Pohl, K.; Wessling, F. (2016). Coordinated Run-Time Adaptation of Variability-Intensive Systems: An Application in Cloud Computing. (TAAS 2016)
- Munoz, Daniel-Jesus; Montenegro, José A.; Pinto, Mónica; Fuentes, Lidia (2018). Energy-aware environments for the development of green applications for cyber-physical systems. (Journal of Systems and Software – Elsevier, 2018)

Murguzur, A.; Truong, H.; Dustdar, S. (2013). Multi-perspective Process Variability: A Case for Smart Green Buildings (Short Paper). (CAiSE Workshops 2013)

Nešić, D.; Nyberg, M. (2016). Multi-view modeling and automated analysis of product line variability in systems engineering. (Journal of Systems and Software – Elsevier, 2016)

Nie, K.; Yue, T.; Ali, S. (2013). Towards a search-based interactive configuration of cyber-physical system product lines. (SPLC 2013)

Nie, Kunming; Yue, Tao; Ali, Shaukat; Zhang, Li; Fan, Zhiqiang (2013). Constraints: The Core of Supporting Automated Product Configuration of Cyber-Physical Systems. (SPLC 2013)

Pohl (2005). Variability Management in Software Product Line Engineering. (Livro – Springer, 2005)

Ramesh, S.; Vogel-Heuser, B.; Chang, W.; Roy, D.; Zhang, L.; Chakraborty, S. (2017). Invited: Specification, Verification and Design of Evolving Automotive Software. (DAC 2017)

Romero, Daniel; Quinton, Clément; Duchien, Laurence; Seinturier, Lionel; Valdez, Carolina (2015). SmartyCo: Managing Cyber-Physical Systems for Smart Environments. (SPLC 2015)

Safdar, Safdar Aqeel; Yue, Tao; Ali, Shaukat; Lu, Hong (2016). Evaluating Variability Modeling Techniques for Supporting Cyber-Physical System Product Line Engineering. (SPLC 2016)

Sascha Julian Oks; Max Jalowski; Michael Lechner; Stefan Mirschberger; Marion Merklein; Birgit Vogel-Heuser; Kathrin M. Möslin. Cyber-Physical Systems in the Context of Industry 4.0: A Review Categorization and Outlook. (Procedia CIRP – Elsevier, 2018)

Sierla, Seppo; O'Halloran, Bryan M.; Nikula, Heikki; Papakonstantinou, Nikolaos; Tumer, Irem Y. (2014). Safety analysis of mechatronic product lines. (Journal of Systems and Software – Elsevier, 2014)

Smirnov, A.; Sandkuhl, K.; Shilov, N.; Telsya, N. (2015). Context variation for service self-contextualization in cyber-physical systems. (CISIS 2015)

Smirnov, A.; Sandkuhl, K.; Shilov, N.; Teslya, N. (2018). Service self-contextualization in cyber-physical systems based on context modeling and context variation. (Future Generation Computer Systems – Elsevier, 2018)

Tenbergen, B.; Daun, M.; Aluko Obe, P.; Brings, J. (2018). View-Centric Context Modeling to Foster the Engineering of Cyber-Physical System Networks. (SPLC 2018)

Thiel, S.; Ferber, S.; Fischer, T.; Hein, A.; Schlick, M. (2001). A case study in applying a product line approach for car periphery supervision systems. (SPLC 2001)

Thiel, S.; Hein, A. (2002). Modeling and using product line variability in automotive systems. (SPLC 2002)

Thiel, Steffen; Hein, Andreas (2002). Systematic Integration of Variability into Product Line Architecture Design. (SPLC 2002)

White, Jules; Clarke, Siobhan; Groba, Christin; Dougherty, Brian; Thompson, Chris; Schmidt, Douglas C. (2010). R&D challenges and solutions for mobile cyber-physical applications and supporting Internet services. (Journal of Internet Services and Applications – Springer, 2010)

Yue, T.; Ali, S.; Lu, H.; Nie, K. (2016). Search-based decision ordering to facilitate product line engineering of cyber-physical systems. (SPLC 2016)

Yue, Tao; Ali, Shaukat; Selic, Bran (2015). Cyber-physical System Product Line Engineering: Comprehensive Domain Analysis and Experience Report. (SPLC 2015)