

## **Origem da Pesquisa**

O projeto é oriundo de pesquisas e orientações relacionadas ao ensino de engenharia de software e programação, áreas que têm ganho destaque no contexto acadêmico e profissional. A pesquisa está vinculada ao mestrado em Computação Aplicada da UTFPR, cuja subárea de Engenharia de Software aborda temas como educação em computação, desenvolvimento de ferramentas educacionais, redução de custos computacionais e financeiros, e melhoria da usabilidade de softwares. A motivação para este estudo surgiu da necessidade de explorar lacunas no ensino de programação, especialmente no que diz respeito à análise de logs acadêmicos e ao uso de ferramentas de captura de eventos durante o processo de aprendizagem.

---

## **Local de Realização**

A pesquisa será realizada no IFAM Campus Boca do Acre, onde o proponente atua como professor EBTT na área de informática. O campus oferece infraestrutura adequada para a coleta de dados e a realização de experimentos, além de contar com um público-alvo diretamente envolvido com o tema da pesquisa.

## **Público-Alvo**

O público alvo da pesquisa serão os estudantes matriculados no curso técnico em informática do IFAM Campus Boca do Acre. Eventualmente, a pesquisa poderá ser estendida para a comunidade local por meio de cursos de programação ofertados pela instituição, ampliando o escopo de coleta de dados e análise.

---

## **Bases Metodológicas**

As bases metodológicas da pesquisa incluem:

1. Revisão Bibliográfica: Leituras de artigos científicos, teses e dissertações disponíveis em repositórios acadêmicos e na biblioteca virtual da UTFPR, com foco em estudos relacionados ao ensino de programação, análise de logs e ferramentas de captura de eventos.
2. Coleta de Dados: Utilização de ferramentas de captura de eventos durante as aulas de programação, registrando ações como criação de variáveis, uso de palavras reservadas, sintaxe e correção de códigos.
3. Análise Quantitativa: Processamento e análise dos logs acadêmicos coletados, com o objetivo de identificar padrões e dificuldades comuns entre os estudantes.

4. Criação de Repositórios: Organização dos dados coletados em um repositório acessível, que poderá ser utilizado por outros pesquisadores e estudantes para futuras pesquisas e aplicações.

---

## **Importância do Estudo**

A pesquisa tem grande relevância acadêmica e prática, pois busca preencher uma lacuna ainda pouco explorada no ensino de programação: a análise de logs acadêmicos. Através da captura e análise de eventos durante o processo de desenvolvimento de códigos, será possível:

- Identificar as principais dificuldades enfrentadas pelos estudantes.
- Propor melhorias no ensino de programação com base em dados quantitativos.
- Criar um dataset de logs acadêmicos que poderá ser compartilhado e utilizado em futuras pesquisas.
- Desenvolver ferramentas educacionais mais eficazes, com foco na usabilidade e na redução de custos computacionais.

Além disso, o estudo contribuirá para a formação de profissionais mais qualificados na área de informática, preparando-os para os desafios do mercado de trabalho.

---

## **Objetivos Gerais**

1. O que pretende: Investigar o processo de aprendizagem de programação por meio da análise de logs acadêmicos, identificando padrões e dificuldades comuns entre os estudantes.
2. Como pretende: Utilizando ferramentas de captura de eventos durante as aulas de programação, processando os dados coletados e organizando-os em um repositório acessível.
3. Por que pretende: Para melhorar o ensino de programação, oferecendo outras características, objetos de estudos baseados em dados quantitativos e criando um recurso (dataset) que possa ser utilizado por outros pesquisadores e educadores.

## **Resumo:**

No ensino de programação, é comum utilizar métricas para avaliar códigos e softwares desenvolvidos durante o aprendizado. A maioria dos métodos atuais foca em análises qualitativas, verificando se o processo é replicável e se está alinhado com as especificações do projeto. No entanto, há uma lacuna na análise quantitativa sobre como os estudantes desenvolvem seus trabalhos, gerando rastros durante a criação dos códigos. Este trabalho propõe

a utilização de logs, que registram atividades de sistemas e usuários, aplicados ao contexto educacional. A ideia é criar ferramentas para capturar, armazenar e sincronizar essas informações em repositórios públicos, utilizando tecnologias como plugins para editores de texto e sistemas de versionamento de código. Embora não seja o foco principal, a ferramenta pode auxiliar docentes a avaliar o desenvolvimento de algoritmos, o tempo gasto nas tarefas, eventos esperados e inesperados, e até identificar possíveis plágios. A análise de logs permite uma avaliação quantitativa e impessoal do processo de aprendizado, indo além da simples entrega da tarefa, e pode contribuir para um ensino de programação mais robusto, ajudando os estudantes a cometer menos erros e a superar dificuldades futuras.

## Introdução:

Enquanto o ensino tradicional prioriza a leitura e a escrita como ferramentas para interpretação e produção de textos, os cursos de programação introduzem uma nova linguagem: os algoritmos. Esses algoritmos, que são sequências lógicas de instruções executáveis por máquinas, representam um tipo de texto único, interpretado tanto por humanos quanto por computadores. Durante o desenvolvimento desses algoritmos, são gerados registros automáticos, conhecidos como logs, que capturam detalhes sobre o processo de criação, execução e depuração do código. Esses registros podem ser transformados em dados valiosos para entender como os estudantes aprendem e interagem com a programação.

Apesar do potencial dos logs para revolucionar o ensino de programação, ainda há desafios significativos na coleta, organização e interpretação desses dados. Muitas vezes, as ferramentas existentes não são adaptadas ao contexto educacional, limitando sua utilidade para professores e alunos. Este trabalho propõe investigar como os logs podem ser integrados de forma mais eficiente ao ensino de programação, explorando métodos inovadores para coletar e analisar esses dados, bem como transformar essas informações em ações práticas e direcionadas.

A relevância dessa abordagem vai além da sala de aula. Ao utilizar logs para monitorar o progresso dos alunos, é possível criar um ciclo de feedback contínuo, onde os instrutores podem identificar padrões de dificuldades, adaptar o conteúdo às necessidades individuais e promover uma aprendizagem mais eficiente. Além disso, a análise de logs pode contribuir para a pesquisa em educação computacional, fornecendo evidências empíricas sobre como os estudantes aprendem programação e quais estratégias pedagógicas são mais eficazes.

## Hipótese

Os resultados provenientes desta pesquisa consistirão em dados de telemetria que poderão gerar novas investigações, conceitos e até mesmo avanços no entendimento pedagógico do ensino de programação. Além disso, diversas subáreas da computação poderão se beneficiar

desses resultados, como ciência de dados, bancos de dados não relacionais e outras áreas interdisciplinares.

Os frutos desta pesquisa incluirão informações detalhadas sobre o uso de computadores e ferramentas computacionais em sala de aula, contribuindo também para campos como pedagogia e psicologia. Adicionalmente, os estudantes envolvidos serão introduzidos a conceitos de telemetria e ao trabalho monitorado por computadores, ampliando seu conhecimento técnico e prático. A utilização de telemetria no contexto educacional representa uma inovação significativa, abrindo caminho para metodologias de ensino mais personalizadas e eficientes.

Essa abordagem não só aprimora o ensino de programação, mas também prepara os estudantes para um mercado de trabalho cada vez mais orientado por dados e automação.

## Objetivos

### Objetivo Geral

O objetivo geral deste trabalho é desenvolver e implementar uma extensão para coleta e análise de logs durante o processo de ensino-aprendizagem de programação, com o intuito de fornecer métricas que auxiliem na identificação de dificuldades dos alunos, na personalização do ensino e no aprimoramento das práticas pedagógicas

### Objetivos Específicos

1. Desenvolver uma extensão para registro de eventos durante as aulas de programação.
2. Implementar o plugin na IDE utilizada na disciplina, com a anuência da instituição e dos estudantes.
3. Coletar, sincronizar e armazenar dados de eventos gerados durante o experimento.
4. Desenvolver um dataset com os dados coletados, organizados por turma, instituição e disciplina.

### Metodologia proposta

Este trabalho adota uma abordagem metodológica composta por duas etapas principais: uma revisão sistemática da literatura e um survey.

#### Revisão Sistemática da Literatura:

A revisão sistemática foi conduzida para investigar como os logs têm sido aplicados no ensino de programação. Foram consultadas cinco bibliotecas virtuais (Scopus, IEEE Xplore, SOL SBC, ACM e WILEY), utilizando combinações de palavras-chave e operadores lógicos. A questão central que guiou a pesquisa foi: "Como os logs têm sido utilizados no ensino de programação?". Além disso, foram definidas sub-questões para explorar aspectos específicos, como objetivos,

ferramentas, métodos e métricas utilizados. Essa etapa visa mapear o estado da arte e identificar lacunas e oportunidades para o desenvolvimento de novas soluções.

#### Survey:

A segunda etapa consiste na aplicação de um survey, cujo formulário contém perguntas relacionadas ao uso de ferramentas de log no apoio ao ensino de programação. As questões abordam temas como:

- O uso de monitoramento no processo de ensino-aprendizagem;
- Como os dados de log podem auxiliar na tomada de decisões pedagógicas;
- O editor de texto ou IDE mais utilizado (informação essencial para o desenvolvimento de um plugin);
- A disposição das instituições em adotar a ferramenta proposta, uma vez disponível.

O survey tem como objetivo coletar percepções e necessidades de professores, alunos e instituições, fornecendo subsídios para o desenvolvimento de uma solução alinhada às demandas reais do contexto educacional.

#### Riscos

Todo projeto computacional voltado ao ensino apresenta possíveis riscos. No caso desta proposta, os principais riscos identificados são:

1. Falha técnica no funcionamento do plugin
  - Possibilidade de o plugin não executar conforme o esperado devido a:
    - Incompatibilidade com versões desatualizadas do editor de texto principal;
    - Interferência de antivírus ou firewalls;
    - Problemas de conexão com o repositório remoto;
    - Erros durante o armazenamento dos logs.
2. Recusa do docente em utilizar a ferramenta
  - O professor responsável pode, por motivos técnicos ou pessoais, optar por não instalar o plugin nos computadores dos estudantes.
3. Negativa do discente em participar da coleta de dados
  - Embora a ferramenta colete apenas eventos relacionados ao ambiente de texto (sem armazenar dados sensíveis ou interferir em outras atividades), o estudante pode optar por não contribuir com os logs.
4. Exposição inadvertida de dados
  - Apesar de o sistema não registrar informações pessoais ou sensíveis, há um risco mínimo de que eventos acadêmicos (como interações no editor) sejam visualizados por outros participantes sem autorização.

## **Benefícios: Logs como Ferramenta Educacional no Ensino de Programação**

O ensino de programação não se limita à transmissão de conhecimentos técnicos, mas abrange também aspectos como licenciamento de código, boas práticas de desenvolvimento e integridade acadêmica. Nesse contexto, o combate ao plágio é uma preocupação central, e os logs emergem como um recurso valioso para apoiar a avaliação discente.

Por meio da análise de padrões de estilo de programação — registrados em um user profile (perfil de usuário) —, é possível identificar discrepâncias na autoria de códigos. Se um trabalho submetido apresenta características incompatíveis com o histórico do estudante (como estrutura, nomenclatura ou lógica), isso pode indicar plágio ou colaboração não autorizada. Além disso, esses dados permitem:

Avaliação personalizada – Acompanhar a evolução individual do discente, detectando dificuldades e adaptando o ensino conforme necessário.

Geração de datasets para pesquisa – Os logs podem ser organizados em tabelas para estudos empíricos sobre metodologias de ensino, estilos de codificação ou eficácia de ferramentas educacionais.

Dessa forma, os logs não só reforçam a originalidade do trabalho, mas também enriquecem o processo pedagógico, fornecendo informações em dados.

## **Metodologia de Análise de Dados**

A análise dos dados coletados pelo plugin será realizada em três etapas principais:

### **1. Coleta e Pré-processamento**

#### **1.1 Registro de Logs:**

Os dados brutos são gerados a partir da interação dos estudantes com o ambiente de programação, capturando:

#### **1.2 Timestamp (data/hora de cada ação);**

##### **1.2.1 Trechos de código editados;**

##### **1.2.2 Padrões de estilo (ex.: nomenclatura de variáveis, estrutura de funções);**

##### **1.2.3 Eventos como compilações, erros e correções.**

#### **1.3 Filtragem e Anonimização:**

Dados sensíveis (como nomes ou matrículas discentes) são substituídos por identificadores únicos, garantindo a privacidade.

### **2. Análise de Autoria e Detecção de Plágio**

#### **2.1 Perfil do estudante (\*User Profile\*):**

- Frequência de commits;

- Uso de estruturas específicas (ex.: loops, funções).

### 3. Geração de Dados Educacionais a partir de eventos

#### 3.1 Dataset para Pesquisa

#### 3.2 Eficácia de metodologias de ensino;

#### 3.3 Correlação entre padrões de codificação e desempenho acadêmico.

### **Desfecho primário:**

O principal resultado esperado é a implementação bem-sucedida de um sistema de coleta e armazenamento de logs das atividades de programação dos estudantes, que:

1. Capture de forma confiável os dados de interação com o ambiente de desenvolvimento
2. Armazene as informações de forma segura e organizada
3. Mantenha a integridade dos dados coletados

### **Desfecho secundário:**

Como resultados complementares, espera-se que:

1. A extensão desenvolvida tenha impacto neutro ou positivo no processo de ensino-aprendizagem
2. A ferramenta opere com mínima interferência no fluxo de trabalho dos usuários
3. Os dados coletados possam ser efetivamente utilizados para análise acadêmica