

# Documentação do projeto JSBridge

Fabio Veiga - 822129856      Gustavo Enzo - 822226174  
Lucas Ota - 822166298

15 de junho de 2024

## Resumo

Este documento descreve o projeto final de Teoria da Computação e Compiladores, incluindo detalhes sobre suas funcionalidades, uso e arquitetura do projeto.

## Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Objetivos</b>	<b>2</b>
<b>3</b>	<b>Funcionalidades Principais</b>	<b>2</b>
<b>4</b>	<b>Usabilidade</b>	<b>3</b>
<b>5</b>	<b>Arquitetura do Projeto</b>	<b>3</b>
<b>6</b>	<b>Conclusão</b>	<b>4</b>

# 1 Introdução

JsBridge implementa um compilador que transforma código escrito em uma linguagem de entrada específica em linguagem de saída, apresentando diferentes níveis de análise. O compilador inclui funcionalidades como análise sintática, geração de linguagem intermediária e conversão para a linguagem final. A interface web permite visualizar e copiar o código em diferentes estágios do processo de compilação.

## 2 Objetivos

Desenvolver um compilador em ambiente web, que permite transformar código escrito em uma linguagem de entrada específica para uma linguagem de saída final. O projeto visa aplicar conceitos de Teoria da Computação e Compiladores, proporcionando uma ferramenta educacional que facilita a compreensão do processo de compilação. A interface web interativa exibe o código em diferentes estágios, desde a análise sintática até a geração da linguagem final, aprimorando a experiência de aprendizado dos usuários.

## 3 Funcionalidades Principais

JSBridge possui várias funcionalidades chave que facilitam o aprendizado e a experimentação:

- **Análise Léxica:** Esta etapa identifica e categoriza os tokens presentes no código fonte, transformando a sequência de caracteres em unidades léxicas significativas.
- **Análise Sintática:** Utilizando gramáticas formais, esta fase verifica a estrutura do código fonte para garantir que ele esteja sintaticamente correto.
- **Análise Semântica:** Verifica a correção semântica do código fonte, assegurando que as instruções façam sentido lógico dentro do contexto do programa.
- **Geração de Código:** Traduz os tokens para linguagem intermediária e então para linguagem final (no caso, `JavaScript`).
- **Ambiente de Teste:** Inclui ferramentas e exemplos que permitem aos usuários testar e validar suas implementações, promovendo um entendimento prático dos conceitos teóricos.

## 4 Usabilidade

JSBridge foi projetado com foco na usabilidade, proporcionando uma interface web intuitiva para facilitar o processo de compilação. Os usuários podem inserir o código fonte e visualizar as etapas de análise léxica, sintática e semântica, bem como a geração de código intermediário e final. A interface interativa permite a alternância entre essas fases, oferecendo uma visualização clara e organizada do fluxo de compilação, o que torna o aprendizado e a utilização mais acessíveis e eficazes para estudantes e desenvolvedores.

## 5 Arquitetura do Projeto

A arquitetura do projeto *JSBridge* é modular e bem organizada, facilitando a manutenção e a extensão do código. A seguir, uma visão geral dos principais componentes:

- **Interface:** O arquivo `editor.js` É responsável por atribuir funções aos elementos visuais do sistema, em adição, a api `taboverride` foi utilizada para implementar um ambiente de texto parecido com uma IDE.
- **Inicialização e Gestão:** O arquivo `main.js` Inicializa o sistema, responsável por obter o input do usuário e chamar os métodos das outras classes.
- **Linguagem:** O arquivo `linguagem.js` armazena informações e expressões aceitas pela linguagem.
- **Análise léxica:** O arquivo `lexer.js` executa a análise léxica do código e retorna, se válido, uma árvore com os tokens.
- **Análise Semântica:** O arquivo `semantic.js` realiza a análise semântica, verificando a coerência lógica do código.
- **Centralização das Análises:** O arquivo `parser.js` centraliza a lógica de análise léxica, sintática e semântica, verificando tokens e estruturas do código fonte.
- **Geração de Código intermediário:** O arquivo `intermediate.js` gera o código intermediário utilizando algoritmos de substituição de termos da cadeia de tokens gerada pelo código do usuário.

- **Geração de Código final:** O arquivo `codigoFinal.js` gera o código final utilizando algoritmos de substituição de termos do código intermediário.

## 6 Conclusão

*JSBridge* é uma excelente ferramenta educacional que oferece uma visão aprofundada do processo de compilação. Com suas funcionalidades abrangentes, foco no aprendizado prático e uma arquitetura bem organizada.