

# RELATÓRIO 2 DE TEORIA DOS COMPILADORES



**Felippe Rocha Lôbo de Abreu - 201765185AC**

**Nathan Toschi Reis - 201865064C**

## **Decisões)**

Optamos por usar o beaver, para mantermos a estrutura jflex anteriormente criada e pela aparente facilidade da gramática apresentada no site oficial.

Principal planejamento foi pensar em como implementar as funcionalidades apresentadas do lang na nossa gramática ao mesmo passo de atualizar nossa estrutura léxica

Após a criação do .grammar decidimos criar exemplos de códigos da linguagem lang em

arquivo .txt e assim desmembrá-los nas gramáticas usando como auxílio as definições de lang.

e assim fomos compilando e executando caso a caso e incrementando as estruturas léxicas e sintáticas

## Como rodar)

- 1) Descompactar o arquivo.zip e abrir o terminal dentro do mesmo.
- 2) Digitar o comando no terminal: “java -cp bin src.Teste Arquivo.txt” dentro do diretório src/

## Problemas enfrentados)

Tivemos uma dificuldade inicial em entender o que deveríamos entregar neste trabalho, e acabamos focando inicialmente em construir uma AST. Tivemos um grande problema para compilar/executar os arquivos, o que causou o atraso na entrega. Depois que resolvemos nosso problema com o java focamos no grammar e em construir seus retornos de acordo com o lang.pdf, mas o atraso já havia comprometido o prazo. Tivemos problemas com algumas partes da gramática como o ELSE que geraram erros na compilação como esse: (Invalid expression: Cannot recover from the syntax error ) e assim não conseguimos garantir a compilação, apesar de termos definidos os terminais e ter construído a gramática de forma aparentemente correta e também gerado os arquivos de parser léxico e sintático.

Decidimos também manter os nomes das variáveis como a do professor, pelo menos no momento. Também não foi implementado a function e data, por conta dos atrasos.

Por fim, estamos tendo uma dificuldade do programa identificar um “char”, e também é algo que estamos focando nossa atenção.