

	<div>Universidade Federal do Ceará – Campus de Russas</div> <div>Compiladores</div>		AV1	AV2	PF
		Trabalho	P1		
		Prova			
Curso:	Disciplina: Compiladores	Datas: data <a href="#">Sigaa[7]</a>			
Professor: Cenez Araújo de Rezende					
Alunos(as):					

## 1 Instruções preliminares:

1. A atividade avaliativa deve ser realizada em equipe de até 4 **alunos**; Você pode formar grupos e agendar acessando e inserindo o nome dos participantes [aqui\[1\]](#);
2. Soluções iguais (de grupos diferentes) possuem prejuízos na pontuação;
3. Forma de Entrega: Projeto e Apresentação dos Detalhes do Trabalho (**Seminário**). **Caso** o grupo **prefira**, pode ser agendado e submetido **link de vídeo** com a apresentação dos **resultados**.
4. A entrega é em:
  1. **Submissão da gramática até a data do [Sigaa\[7\]](#);**
  2. **Apresentações de resultados: cronograma no Sigaa e Planilha de agendamento;**
5. A equipe é inteiramente responsável por discorrer sobre os detalhes de projeto, dentro dos elementos estabelecidos no **item 2**.

## 2 Projeto de Compilador (Parte 1)

**Problema:** compiladores são desenvolvidos em etapas. Como vimos em nossas aulas, são várias as etapas. Contudo, elas podem ser divididas em duas partes: Front-end e Back-end, formando o processo de compilação, composto de análise e síntese. Na análise, o compilador busca entender o código fonte, verificando se está de acordo com a linguagem, para que seja possível construir uma estrutura intermediária. Na síntese, o compilador constrói o código objeto a partir da representação intermediária. Vamos, neste momento, focar na análise. Para isso, precisamos definir como seria a linguagem que queremos especificar. Assim, partimos de uma gramática: este é o primeiro passo. Suponha agora que a gramática esteja definida. Há várias formas de testá-la. Ou seja, verificar se um código (palavra) pertence a ela. Podemos fazer isso usando autômatos com pilha, a ferramenta [Jflap\[2\]](#), ou literalmente construindo elementos recursivos (ou não) relacionados à gramática. Nosso problema então é definir o passo inicial, seguindo as especificações seguintes.

### 2.1 Especificação da Atividade

- Definir uma gramática de linguagem, visando linguagem de programação. Aqui você pode buscar ideias de gramáticas básicas, como operações matemáticas, e ir adicionando recursos. Reflita bastante sobre o que gostaria de ter (sintaticamente) em uma linguagem de programação. Pode-se observar gramáticas reais disponíveis na internet, buscando entendê-las para alterar e colocar sua assinatura, como projetista de linguagem. Veja um exemplo [aqui\[6\]](#);
- Documentar os detalhes da linguagem. O que o grupo pensou sobre as variáveis (da gramática), que irão fazer produções?
- Utilizar alguma ferramenta que permita testar características da gramática, como [Jflap\[2\]](#), nosso autômato com [pilha\[5\]](#) estudado na disciplina de LFA, ou ainda ferramentas geradoras como “yacc” ou “[Bison](#)”[3];
- Como ficaria a análise léxica no nosso projeto em andamento, cujos códigos temos trabalhado no laboratório? Que tipo de gramática está usando? Problemas ou virtudes?
- Apresentar os resultados do projeto em um breve seminário;

- Inserir no [Sigaa\[7\]](#) os detalhes do estudo, incluindo a gramática analisada ou desenvolvida.

### 3. Critérios Principais na Aferição do Andamento do Trabalho:

1. Clareza na especificação e apresentação do projeto;
2. Trabalho em equipe, uma vez que potencialmente é comum no desenvolvimento e projeto de software;
3. Testes diversos: pode-se utilizar algum tipo de ferramenta, como [Jflap\[2\]](#), autômato com pilha[5], [Bison\[3\]](#), [BnfPlayGround\[4\]](#), nosso projeto em Python, etc. No Caso do [BnfPlayGround\[4\]](#), você pode encontrar algumas limitações da ferramenta;
4. Apresentação dentro do prazo.

### 4. Links de referência:

- [1] [https://docs.google.com/spreadsheets/d/19sclZLV4N9KF3MFhuS8QJg\\_zI-D9PNUwWPVI8mGflcw/edit?gid=0#gid=0](https://docs.google.com/spreadsheets/d/19sclZLV4N9KF3MFhuS8QJg_zI-D9PNUwWPVI8mGflcw/edit?gid=0#gid=0)
- [2] <https://www.jflap.org/>
- [3] <https://www.gnu.org/software/bison/>
- [4] <https://bnfplayground.pauliankline.com/>
- [5] <https://replit.com/@CenezAraujo/PDA>
- [6] [https://drive.google.com/file/d/1wBXk4c6FC6n4\\_PgZPH4IzAKw2zsSpjkm/view?usp=sharing](https://drive.google.com/file/d/1wBXk4c6FC6n4_PgZPH4IzAKw2zsSpjkm/view?usp=sharing)
- [7] <https://si3.ufc.br/sigaa/verTelaLogin.do>