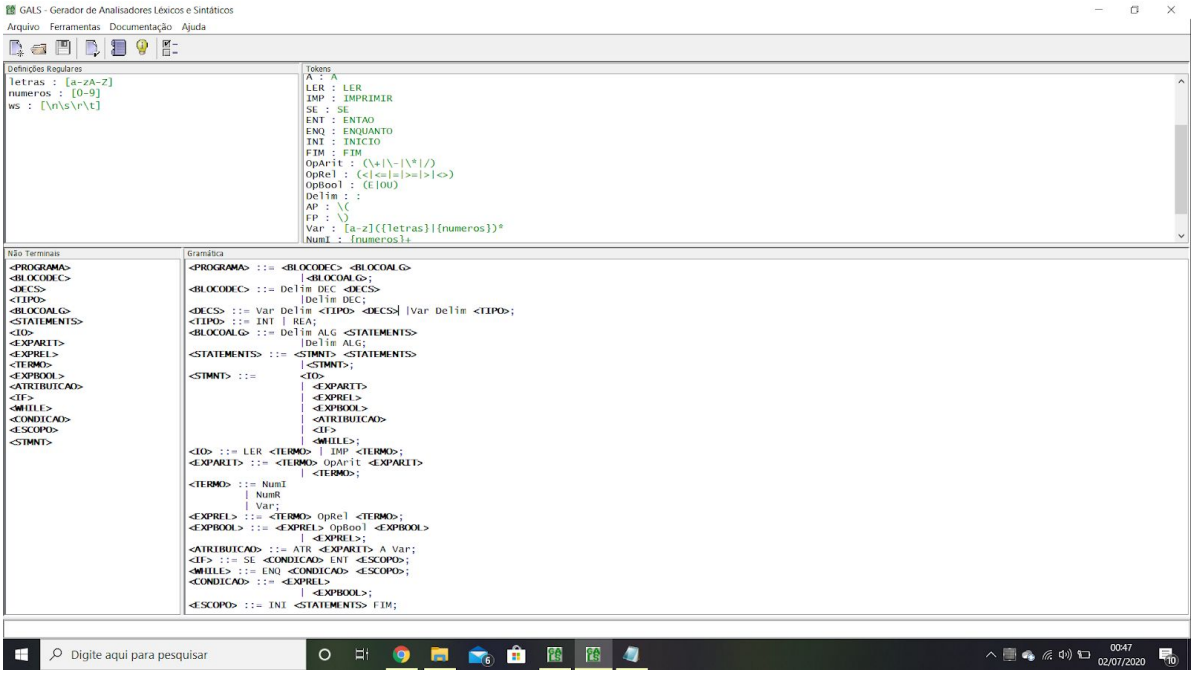
REOblazer (4) - Atividade 2

Mateus Carvalho Gonçalves - 201810245

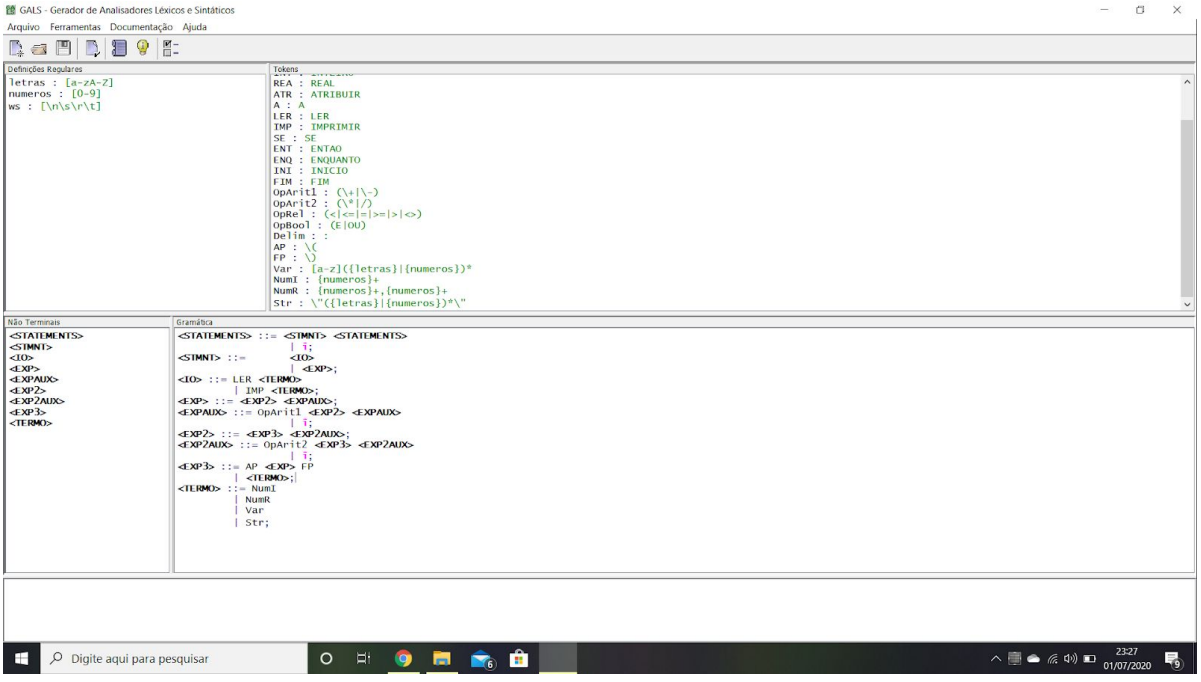
Primeiramente, este é o último analisador produzido para a linguagem ALGUMA:



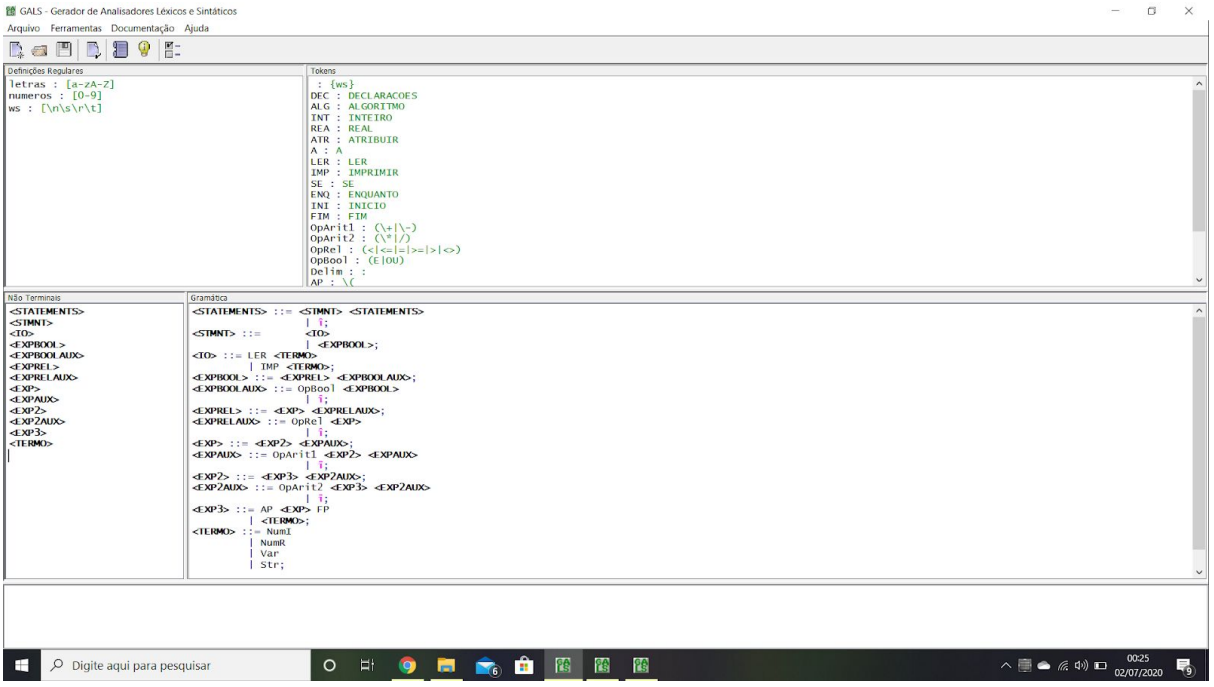
Para transformar em uma gramática LL(1), foi adotada uma tática de começar pelos statements, mais especificamente pelas expressões aritmética, e se fazer de forma gradual.

Algumas pequenas mudanças foram feitas inicialmente para abranger melhor a gramática (e.g. <TERMO> ::= Str e exclusão do token ENT, uma vez que foi definido escopo INI/FIM).

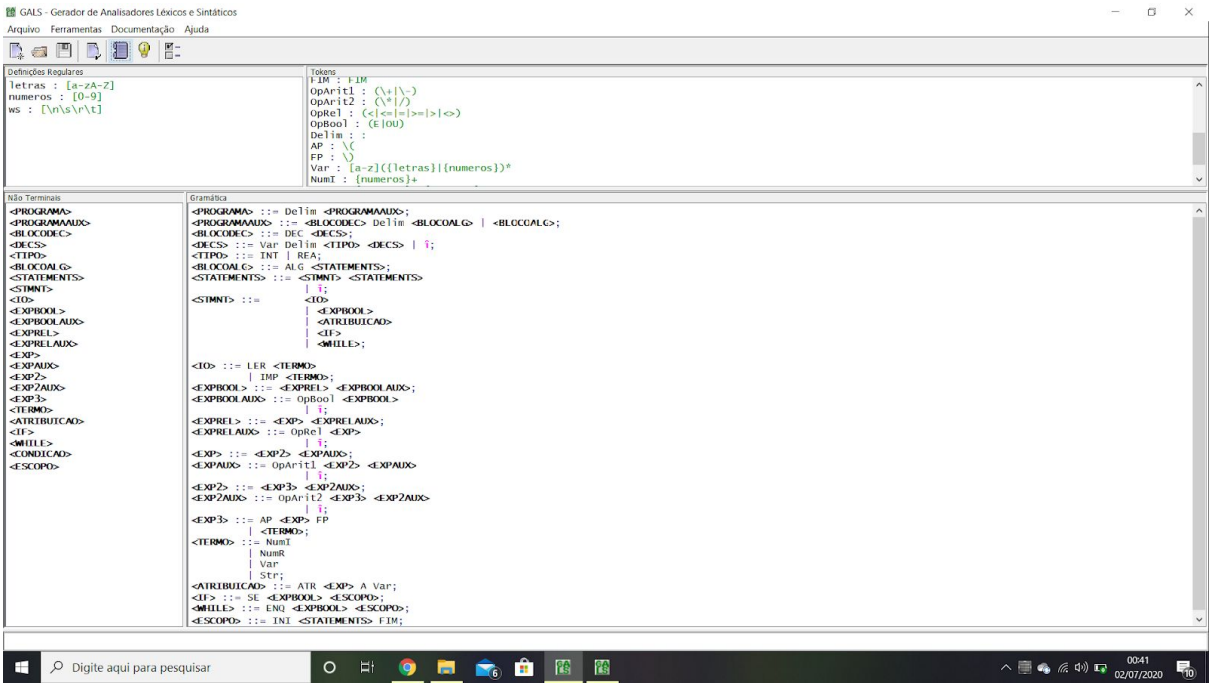
Com base em um exemplo de aula, foi adotado a precedência de operadores em expressões aritméticas, e para isso, a definição dos tokens foi mudada separando {+, -} de {*, /}.



Depois foram adicionadas as expressões relacionais e booleanas. Um fato que achei curioso foi a organização das expressões, todas contidas umas nas outras, para se ter um único termo é necessário passar por todas as produções de expressões desde booleanas até aritméticas e suas auxiliares, de modo a evitar ambiguidade e recursão a esquerda.

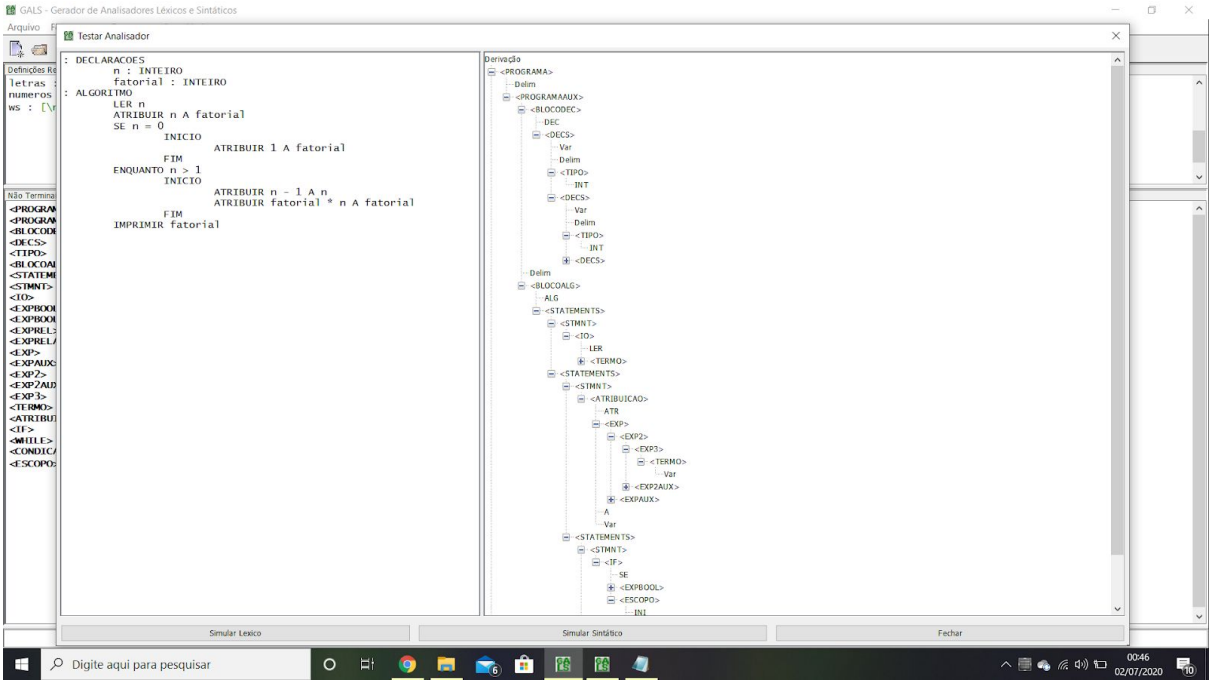


Para finalizar foram organizados os outros statements e blocos do código.



A gramática sofreu inúmeras mudanças para se tornar LL(1), mas pelo caráter trabalhoso da tarefa foi escolhido aceitar transições vazias na gramática. Um teste com um dos algoritmos

feitos no início da disciplina foi rodado para validar a solução, e foi aceito com sucesso sem nenhum warning de ambiguidade.



Ainda que poderiam ser feitas várias melhorias no analisador, a solução já abrange de forma geral todas as regras da linguagem ALGUMA que foram apresentadas nos slides no início do curso de Compiladores.