

# Temas dos Seminários

---

Nos seminários cada grupo deverá primeiramente introduzir, e explicar os conceitos do seu tema, e, posteriormente o relacionar com alguma das partes dos trabalhos práticos desenvolvidos, quando aplicável.

## **G1 - Diferenças entre linguagens Interpretadas e Compiladas**

Diferença entre execução direta e geração de código.

Onde o trabalho implementado se encaixa.

Exemplos, vantagens e desvantagens de linguagens, compiladas, interpretadas e híbridas

## **G5 - Diferenças entre gramáticas LL e LR**

Diferença entre abordagens top-down (LL) e bottom-up (LR).

Por que usamos parsing recursivo descendente e o que torna uma gramática não LL(1)?

Vantagens e limitações de cada abordagem.

## **G7 - Tabela de Símbolos e Análise Semântica**

O que é uma tabela de símbolos?

Como a análise semântica usa essa estrutura para verificar escopo, declaração, tipo, uso de variáveis?

Exemplos de erros semânticos comuns.

## **G8 - Códigos de três endereços e representação intermediária**

O que são, por que usar, exemplos típicos.

Papel como ponte para otimização e geração de código.

Explicar como a dependência de variáveis se relaciona com o problema de coloração de grafos.

## **G2 - Otimização de código intermediário**

Em qual parte das etapas da compilação ocorre a otimização?

O que é otimizar código? Por quê?

Tipos: constantes, dead code, strength reduction, etc.

## **G3 - Tratamento e recuperação de erros**

Diferença entre detectar e recuperar.

Como lidaram com erros nas diferentes fases.

Estratégias mais avançadas para compiladores robustos.

## **G4 - Código Fonte, Código Objeto e Código de Máquina**

Visão da cadeia completa até o executável.

Ferramentas como assemblers e linkers.

Como múltiplos arquivos são unidos.

## **G6 - Ambiente em tempo de execução**

Como variáveis e estruturas de controle são gerenciadas em tempo de execução?

Pilha, heap, chamadas de função.