

# Diário de Bordo (Simulado)

---

Período: Semana 1 e Semana 2

Integrantes: João Zanini, Carlos Vital, Felipe Laskos, João Manoel, Cayo Vinícius.

## Semana 1

### Planejamento Inicial

- Definição das estruturas: lista simples para estudantes, lista duplamente encadeada circular para ocorrências.
- Decisão por nó sentinela para simplificar inserções e remoções e permitir navegação bidirecional.
- Definição dos arquivos e módulos.

### Tarefas e Responsáveis

- João Zanini: Arquitetura das listas e criação/limpeza com sentinela (`lista_criar`, `lista_destruir`), acesso direto às extremidades (`lista_primeiro`, `lista_ultimo`) e busca de estudante (`estudantes_buscar`).
- Carlos Vital: Núcleo do cadastro de estudantes (`estudantes_init`, `estudantes_adicionar`, `estudantes_listar`).
- Felipe Laskos: Integração e gerenciamento de ocorrências (`registrar_ocorrencia`, `listar_ocorrencias_por_ra`, `remover_ocorrencia`, `filtrar_por_tipo`) e utilidades de data (`parse_data`, `data_to_string`).
- João Manoel: Ordenação por data e regra de comparação (`lista_inserir_ordenado`, `comparar_datas`, helper `eh_mais_recente`).
- Cayo Vinícius: Remoção/contagem na lista base (`lista_remover_por_data`, `lista_contar_por_tipo`) e destruição de estudantes (`estudantes_destruir`).

### Dificuldades

- Ordenação por data descendente: resolvido criando função de comparação e laço de inserção.
- Manter código organizado: divisão clara por módulos e comentários de autoria.

### Código Comentado (Exemplo)

Inserção ordenada de ocorrência (`lista_inserir_ordenado`):

```
while (atual != s && !eh_mais_recente(data, atual->data)) {
    atual = atual->next;
}
// Inserir antes de 'atual'
novo->next = atual;
novo->prev = atual->prev;
atual->prev->next = novo;
atual->prev = novo;
```

Explicação: percorremos até achar o primeiro nó cuja data seja mais antiga que a nova (ou sentinela). Inserimos o novo nó imediatamente antes, preservando a ordem decrescente (mais recentes primeiro).

## Semana 2

### Avanços

- Implementação do menu interativo em `main.c`.
- Navegação sequencial para frente e para trás das ocorrências.
- Função de remoção por RA + data concluída.
- Filtro por tipo consolidado.

### Testes Manuais

1. Cadastrar estudante RA 1.
2. Registrar 3 ocorrências em datas diferentes.
3. Listar por RA e navegar (n/p) confirmando a ordem.
4. Remover ocorrência intermediária (data X) e revalidar lista.
5. Filtrar por tipo 'Advertencia'.