1.2 . Justifique a escolha dos tipos de dados.

Índice composto em (pk\_usuario, pk\_ativo, DATETIME):

* Os nomes das chaves foram melhorados para maior legibilidade
* (ex: pk\_usuario em vez de id\_usuario).
  + pk\_usuario: Permite a filtragem rápida por id\_usuario.
  + pk\_ativo: Permite a filtragem rápida por id\_ativo.
* Adicionados prefixos para tipos não especificados (dt\_ para DATETIME e tp\_ para ENUM) para manter a consistência do padrão.
* (ex: dt\_cotacao em vez de data\_hora\_cotacao)
* (ex: tp\_operacao em vez de tipo\_operacao)
  + DATETIME: Permite a filtragem eficiente por data
* Padronização de Prefixos:
  + pk\_ para Chaves Primárias.
  + fk\_ para Chaves Estrangeiras.
  + ds\_ para campos de texto (VARCHAR).
  + vl\_ para campos de valor (DECIMAL).
  + nr\_ para campos numéricos inteiros (INT).

**2. Índices e Performance**

**2.1.** Proponha e justifique um ou mais índices para essa consulta.

SELECT define as colunas que você quer ver.

FROM especifica a tabela de onde os dados virão.

A cláusula WHERE para restringir os resultados.

JOIN é essencial para ligar tabelas.

ORDER BY como a consulta ordena os resultados.

DESC pode ajudar o otimizador a usar o índice de forma mais direta para ordenações decrescentes.