

HASH

```
CREATE INDEX idxHashEmail ON Usuario USING hash (email);
```

Consulta que se beneficiaria do Index.

```
SELECT IdUsuario, Nome, Senha FROM Usuario  
WHERE email = 'carlos.monteiro@email.com';
```

Considerando que quando um usuário for fazer login no sistema escolar ele precisará informar seu email, uma tabela hash para a coluna email seria útil pois um índice do tipo Hash é projetado para otimizar pesquisas de igualdade exata como no caso de email.

GIN

```
CREATE INDEX idxMensagemConteudoPalavrat ON Mensagem USING GIN  
(to_tsvector('portuguese', Conteudo));
```

Consulta que se beneficiaria do Index.

```
SELECT * FROM Mensagem  
WHERE to_tsvector('portuguese', Conteudo) @@ to_tsquery('portuguese', aula');
```

Um índice GIN é projetado para lidar eficientemente com consultas em colunas que contêm valores compostos ou múltiplos valores em uma única coluna, com esse índice podemos acelerar a busca em textos. No caso de Mensagens podemos encontrar de forma mais eficiente palavras mencionadas.

BTREE

```
CREATE INDEX IdxUsuarioNomeSobrenome ON Usuario (Nome, Sobrenome);
```

```
SELECT IdUsuario, Nome, Sobrenome, email FROM Usuario  
WHERE Nome = 'Rafael' AND Sobrenome = 'Martins';
```

Usando o índice B-tree, o otimizador de consultas do PostgreSQL pode usá-lo para encontrar rapidamente os registros que correspondem à sua cláusula WHERE, por ser ordenado, permite que o PostgreSQL vá diretamente para a parte do índice onde os nomes começam com 'Rafael'. Sem o índice precisaria realizar um scan sequência em toda a tabela Usuario.