

1. Iniciamos um processo de transição no curso, falamos até o momento somente de xP, em curto espaço de tempo vamos falar do Scrum. Em alguns momentos vou fazer algumas provocações. Segue a primeira delas:

O texto apresentado no link abaixo defende que "Refinement é um termo novo na boa e “velha” Engenharia de Software".

[<https://engenhariasoftware.wordpress.com/2021/06/06/refinement-scrum/>]

Faça a leitura do texto completo (3 minutos) e responda: Você concorda que "Refinement é um termo novo na boa e “velha” Engenharia de Software"?

Sim ou Não?

Justifique rapidamente sua resposta.

- Resposta
 - sim concordo, refinement é um termo novo na boa e velha engenharia de software.
 - o texto apresenta um termo utilizado por metodologias ágeis em uma etapa da construção do software, sendo essa a de priorização das atividades
 - anteriormente já existia essa etapa de priorizar e componentizar, no entanto as novas filosofias de desenvolvimento ágil trouxeram termos novos para processos antigos.
- Vamos aplicar o Refactoring e Código Coletivo.

Solicite um amigo o código fonte de um programa. Verifique se o código enviado a você é simples e fácil de entender. Verifique se existem uma padronização na nomenclatura das variáveis, um organização do programa em rotinas, se os parâmetros das rotinas estão coerentes.

Após analisar me informe rapidamente se o código necessita ou não de Refactoring.

- Resposta
 - a seguir será apresentado o código enviado em c#

```
[Authorize]
[HttpPost]
public async Task<IActionResult> PostContagem(ContagemDeEstoque contagem, string database)
{
    string strSql = string.Empty;
    string OBS = contagem.Observacao == "" ? "NULL" : $"{contagem.Observacao}";
    await using (NpgsqlConnection conn = ExternalConnection.Conn(database))
    {
        conn.Open();
        strSql += $" INSERT INTO inventario_barras_cabecalho(data_hora, codpto, obs)";
        strSql += $" VALUES(NOW(),{contagem.CodigoPonto},{OBS});";
        strSql += $" SELECT MAX(codigo) AS codigo FROM inventario_barras_cabecalho;";
        NpgsqlCommand command1 = new NpgsqlCommand(strSql, conn);
        NpgsqlDataReader dr = command1.ExecuteReader();
        int codCabecalho = 0;
        while (dr.Read())
        {
            codCabecalho = dr.GetInt32(dr.GetOrdinal("codigo"));
        }
        dr.Close();
        strSql = "";
        foreach (CodigoDeBarraDTO item in contagem.CodigosDeBarra)
        {
            strSql += $" INSERT INTO inventario_barras_itens(cod_cabecalho, data_hora, cod_barras)";
            strSql += $" VALUES({codCabecalho},NOW(),'{item.Descricao}');";
        }
        NpgsqlCommand comando2 = new NpgsqlCommand(strSql, conn);
        comando2.ExecuteNonQuery();
    }
    return CreatedAtAction(nameof(PostContagem), null);
}
```

- Após analisar o código podemos fazer algumas mudanças
 1. alterar strSql de string para system.Text.StringBuilder para aumentar a performance
 2. seguir padrao de nomenclatura de variáveis, ao invés de OBS usar observacoes
 3. usar para validação String.IsNullOrEmpty para ter uma validação mais abrangente
 4. usar dollar para informar strings ao invés de aspas simples, isso é o mínimo para evitar sql injection

```
[Authorize]
[HttpPost]
public async Task<IActionResult> PostContagem(ContagemDeEstoque contagem, string database)
{
    var strSql = new System.Text.StringBuilder();
    string observacoes = String.IsNullOrEmpty(contagem.Observacao) ? "NULL" : $"${contagem.Observacao}$";
    await using (NpgsqlConnection conn = ExternalConnection.Conn(database))
    {
        conn.Open();
        strSql.AppendLine($" INSERT INTO inventario_barras_cabecalho(data_hora, codpto, obs)");
        strSql.AppendLine($" VALUES(NOW(),{contagem.CodigoPonto},{observacoes});");
        strSql.AppendLine($" SELECT MAX(codigo) AS codigo FROM inventario_barras_cabecalho;");
        NpgsqlCommand command1 = new NpgsqlCommand(strSql.ToString(), conn);
        NpgsqlDataReader dr = command1.ExecuteReader();
        int codCabecalho = 0;
        while (dr.Read())
        {
            codCabecalho = dr.GetInt32(dr.GetOrdinal("codigo"));
        }
        dr.Close();
        strSql = new System.Text.StringBuilder();
        strSql.AppendLine($" INSERT INTO inventario_barras_itens(cod_cabecalho, data_hora, cod_barras) VALUES ");
        foreach (CodigoDeBarraDTO item in contagem.CodigosDeBarra)
        {
            strSql.AppendLine($" ({codCabecalho},NOW(),$desc${item.Descricao}$desc$),");
        }

        NpgsqlCommand comando2 = new NpgsqlCommand(strSql.ToString(), conn);
    }
}
```

```
        comando2.ExecuteNonQuery();
    }
    return CreatedAtAction(nameof(PostContagem), null);
}
```

- O código poderia ser refatorado em mais vezes ainda quebrando em mais métodos. outra coisa que poderia ser feita é colocar o try catch.