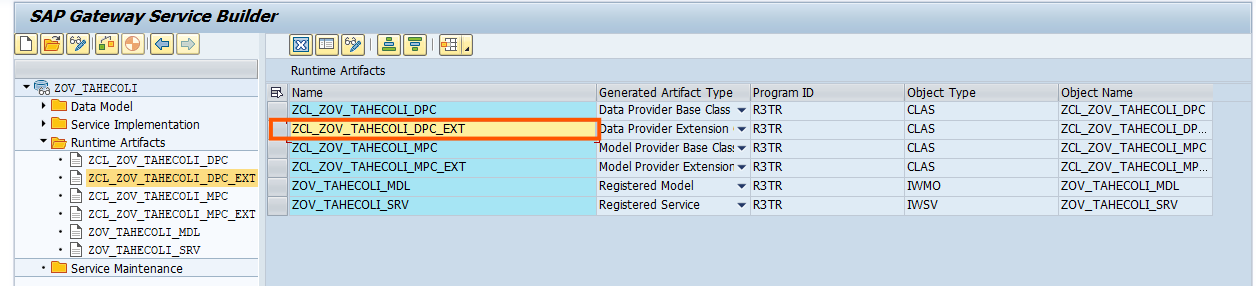
# Implementação back-end ABAP

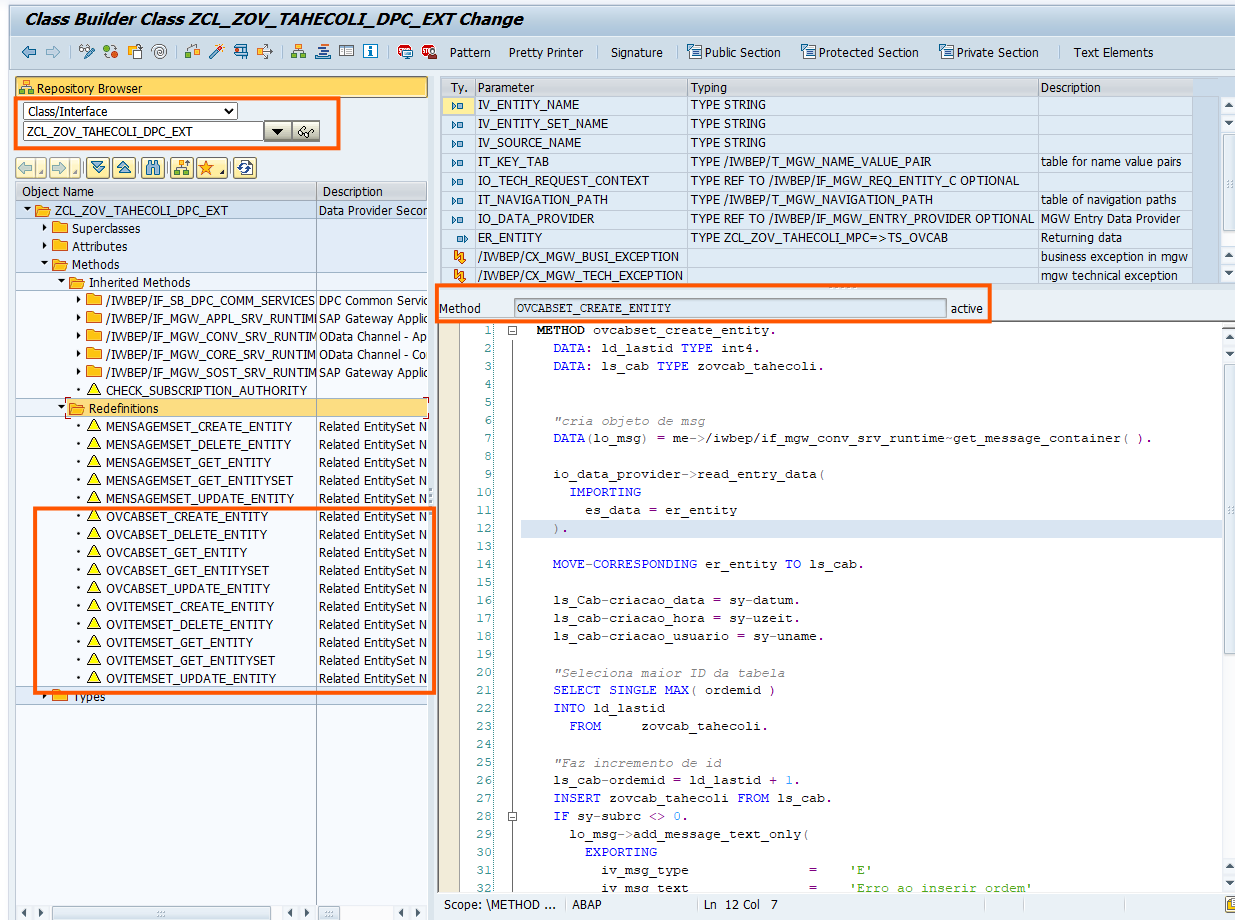
**Aluno:** Taglys Henrique Costacurta de Oliveira  
**RGM:** 30886406  
**Disciplina:** Projeto Integrador Transdisciplinar em Engenharia de Software II  
**Tema:** Sistema de Ordem de Vendas em ABAP **Etapa:** 3 Implementação back-end ABAP   
**Repositório GitHub:** <https://github.com/taglyscostacurta/servico-odata-backend-abap>

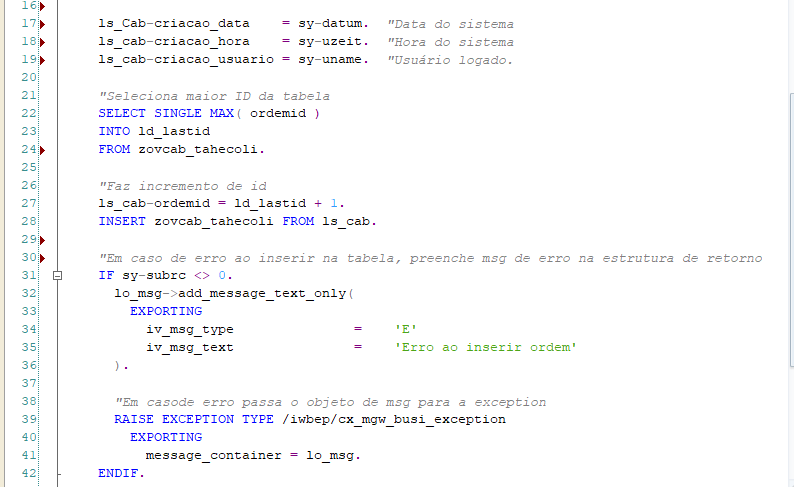
Depois de criar o serviço Odata, o sistema gera automaticamente as classes padrão e de extensão para implementar a logica de negocio.

A classe principal utilizada foi ZCL\_ZOV\_TAHECOLL\_DPC\_EXT, que é responsável por receber as requisições do Odata e executar as operações CRUD nas tabelas ZOVCAB\_TAHECOLI e ZOVITEM\_TAHECOLI.

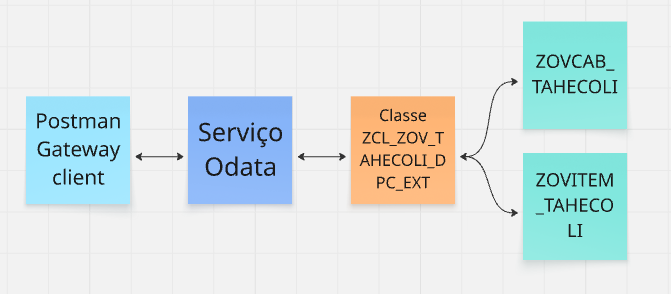


Essa classe tem metodos que podemos **redefinir** e implementar nossa logica.  
Isso para cada função do CRUD e as entidades criadas no serviço Odata.



Neste método OVCABSET\_CREATE\_ENTITY implementei uma logica para que os dados que chegam via serviço odata sejam inseridos na tabela criada anteriormente, nesse caso a tabela zovcab\_tahecoli, antes da inserção atribui usuario que esta logado, data, e hora para que possamos acompanhar as alterações e quem a fez.  
Depois faz uma busca na propria tabela com o campo chave mais alto e utiliza como parametro para fazer um autoincremento de **ordemid**  


**O mesmo padrao de implementaçao é aplicado nos outros metodos do CRUD, (GET\_ENTITY, UPDATE\_ENTITY, DELETE\_ENTITY) e para as demais entidades do serviço, como ovitemset, para que o ciclo de vida das ordens de venda e seus itens seja controlado via serviço odata.**

  
Fluxograma feito no miro