|  |
| --- |
| **Introdução ao Processo** |
| **Objetivo:** Criar um evento a partir da criação de um cliente seguindo os requisitos destacados abaixo **Tipo do Registro:** N/A **Objetos:** Account, Event |
| **Requisitos** |
| 1. Criar Evento na Criação de Clientes **Premium** com faturamento **acima de $1.000.000,00**  2. Precisa ver a Trigger e a chamada para Trigger Handler  3. Design Pattern:   * FixtureFactory * Repository * TriggerHandler * Filter * Enricher   4. Classes com 100% de cobertura |
| **Cenários** |
| **(1)**  *//given*  que um cliente (Account) com (Category\_\_c) "Premium" e Receita Anual de 600.000 (600 mil)  *//when*  inserido, ou seja, criado  *//then*  *Nenhum evento criado*  ***(2)***  *//given*  que um cliente (Account) com (Category\_\_c) "Standard"  *//when*  inserido, ou seja, criado  *//then*  *Nenhum evento criado*  ***(3)***  *//given*  que um cliente (Account) com (Category\_\_c) "Premium" e Receita Anual acima de 1.000.000 (1 million)  *//when*  inserido, ou seja, criado  *//then*  novo Evento *(descrito abaixo)* |
| **Objeto Evento** |
| Description = 'Apresentar o portfólio de Produtos para o novo cliente com enfoque em nossa carteira Premium'  Subject = 'Apresentação institucional dos produtos Premium'  StartDateTime = Agendar para 1 dia após a data de criação do Cliente  EndDateTime = Deverá ter uma hora de duração  WhatId = Id do cliente em questão  ContactType\_\_C (de Activity) = 'Telefônico' |
| **Campos Personalizados** |
| **Campo:** Category (Category\_\_c)  **Objeto:** Account  **Tipo:** Picklist  **Valores:** Standard, Gold, Silver, Premium  **Campo:** Contact Type (ContactType\_\_c) **Objeto:** Activity  **Valores:** Standard, Gold, Silver, Premium **Regra:** Visível apenas para Event |
| **Trigger** |
| • Criação de Trigger para o objeto Account com parâmetro (after insert)  • Utilizar Pattern TriggerHandler  - AccountTrigger terá uma linha:  new LeadTH().run(); |
| **TriggerHandler** |
| • AccountTH (base)  List<Account> newAccounts;  public AccountTH() {  this ((List<Account>) Trigger.new)}  public AccountTH (List<Account> newAccounts) {  this.newAccounts = newAccounts;  }  override  public void afterInsert() {  //todo o código entra aqui  } |
| **TriggerFilter** |
| • AccountFilter |
| **TriggerEnricher** |
| • AccountEnricher |
| **FixtureFactory** |
| AccountFixtureFactory ()   * createAccount (parâmetros)   :aqui retorna createAccount(newAccount(parâmetros))   * newAccount (parâmetros)   :aqui retorna uma nova account com os campos obrigatórios   * createAccount (Account account)   : aqui vai o insert e return account  EventFixtureFactory   * createEvent (parâmetros)   :aqui retorna createEvent(newEvent(parâmetros))   * newEvent (parâmetros)   :aqui faz a declaração de uma nova event com os campos obrigatórios   * createEvent ()   :aqui vai o insert e return event |
| **Classes Mapeadas** |
| * AccountEventEnricher * AccountFilter * AccountFixtureFactory * AccountRepository * AccountRepositoryTest * AccountTH * EventFixtureFactory * EventRepository * EventRepositoryTest * TriggerHandler *@author kevinohara80* * TriggerHandlerTest *@author kevinohara80* |
| **Triggers Mapeadas** |
| * AccountTrigger |