# **Kit de Dados - Projeto de Dashboard (Cannoli + FECAP)**

## 1. Visão Geral do projeto

O objetivo deste projeto é permitir que alunos desenvolvam um dashboard interativo para análise de campanhas de marketing, perfil de clientes e pedidos realizados em uma rede de estabelecimentos.  
  
A aplicação deve gerar análises, cruzamentos e indicadores que auxiliem na visualização de padrões e oportunidades de negócio. Os dados foram anonimizados para garantir a privacidade dos clientes reais.

## 2. Exemplo de Dados (PostgreSQL)

Segue nas próximas páginas as interfaces de cada interface nos bancos disponíveis, com seus contextos.

Consulte os ‘JSON’ enviados para exemplos.

**2.1 CampaignQueue** (Interface utilizada para salvar todos os agendamentos e estados de mensagens) ->

id!: string; // Identificação do registro no banco

jobId!: string; // O mesmo ID do trabalho agendado (disparo)

campaignId!: string; // ID da campanha

storeId!: string; // Identificação daloja

storeInstanceId!: string; // ID da instância do WhatsApp

customerId!: string; // ID do cliente que recebeu a mensagem

phoneNumber!: string; // Telefone do cliente

scheduledAt!: Date; // Para quando o disparo foi agendado

sendAt!: Date; // Data de quando a mensagem foi enviada

status!: CampaignQueueStatus; // Enum – Qual o estado desse disparo

message!: string; // Texto da mensagem enviada

response?: string; // Possível resposta do cliente

createdAt!: Date; // Quando esse registro foi criado

createdBy!: string; // Por quem esse registro foi criado

updatedAt?: Date; // Quando esse registro foi atualizado

updatedBy?: string; // Por quem esse registro foi atualizado

**2.2 CampaignQueueStatus** (Enum (enumerator) utilizado na interface ‘CampaignQueue’) ->

Scheduled = 1,

Send = 2,

Received = 3,

Read = 4,

Deleted = 5,

Pending = 6,

**2.3 Campaign** ->

id!: string; // Identificador da campanha

segmentId?: string; // Identificador do segmento

templateId!: string; // Identificador do template

storeId!: string; // Identificador da loja

name!: string; // Nome da campanha

description?: string; // Descrição da campanha

badge?: string; // Distintivo – O que ela é, exemplo: migration (migração), loyalty (fidelização), consumption (consumo)

type!: CampaignType; // Tipo da Capanha – (1 ou 2, Promocional ou Institucional)

status!: CampaignStatus; // Estado da campanha (Draft = 1 (rascunho), Canceled = 2 (cancelado), Published = 3 (publicado), Completed = 4 (completado),

isDefault!: boolean; // Se a campanha é padrão do sistema ou customizada

createdAt!: Date;

createdBy!: string;

updatedAt?: Date;

updatedBy?: string;

segment?: segmentDto; // Objeto do segmento utilizado na campanha

template?: templateDto; // Objeto do template usado na campanha

store?: storeDto; // Objeto da loja utilizado na campanha

campaignTrigger?: campaignTriggerDto[]; // Lista de disparador (por padrão só haverá 1 índice ([0]), mas foi feito dessa forma para implementações futuras)

**2.4 Customer** ->

id!: string; // Identificador do cliente

name!: string; // Nome do cliente

taxId!: string; // Documento CPF ou CNPJ

gender?: string; // Gênero

dateOfBirth?: Date; // Data de nascimento

status!: CustomerStatus; // Estado – (Active = 1, Inactive = 2)

externalCode?: string; // Código externo da origem da informação

isEnriched?: boolean; // Se as informações do cliente foram enriquecidas (telefone, e-mail)

enrichedAt?: Date; // Quando foi enriquecido

enrichedBy?: string; // Usuário que solicitou o enriquecimento

createdAt!: Date; // Data da criação do registro

createdBy!: string; // Quem registrou a informação

updatedAt?: Date; // Quando o registro foi atualizado

updatedBy?: string; // Quem atualizou o registro

customerAddress?: customerAddressDto[]; // Lista de endereços

customerPhone?: customerPhoneDto[]; // Lista de telefones

customerEmail?: customerEmailDto[]; // Lista de e-mails

segmentCustomer?: segmentCustomerDto[]; // Lista de segmentos que esse cliente faz parte

customerStore?: customerStoreDto[]; // Lista de lojas que esse cliente consome

## 3. Exemplo de Dados (Firebase)

\*\*Nota\*\*: os dados fornecidos são completamente anonimizados.

**3.1 Order** ->

actions: IOrderActionV3[]; // Lista de ações feitas no pedido

benefits: IOrderBenefits[]; // Lista de promoções (cupons utilizados)

cancellation: IOrderCancellationV3; // Objeto de cancelamento do pedido

companyId: string; // Identificação da loja

containerId: string; // Identificação da marca

createdAt: Date; // Data de criação do pedido

customer: CannoliIOrderCustomerV3; // Informações do cliente

deliverer: IOrderDelivererV3; // Informações do entregador (se for pedido ‘delivery’)

delivery: IOrderDeliveryV3; // Informações de entrega (se for pedido ‘delivery’)

displayId: string; // Código curto de exibição

engineId: string; // Identificador do motor de venda

engineName: string; // Nome do motor de venda

engineType: EEngineType; // Tipo do motor de venda

extraInfo: string; // Informações extras

id: string; // Identificador do pedido

indoor: IOrderIndoorV3; // Objeto de informações do pedido dentro do estabelecimento (se o pedido for ‘indoor’)

integrated: boolean; // Se o pedido foi integrado com parceiro externo

integrationId: number; // Identificador da integração

isTest: boolean; // Se é um pedido de testes ou não

items: IOrderItem[]; // Itens (produtos) do pedido

merchant: IOrderMerchantV3; // Objeto de identificação da loja

metadata?: any; // Informações técnicas adicionais

orderTiming: CannoliEOrderV3Timing; // Para quando o pedido foi feito – SCHEDULED ou IMMEDIATE

orderType: CannoliEOrderV3Type; // Tipo do pedido – DELIVERY, INDOOR, TAKEOUT

payments: IOrderPaymentV3; // Informativo de pagamentos

preparationTime: number; // Tempo de preparo em minutos

reference: string; // Referência externa do pedido

salesChannel: CannoliEOrderV3SalesChannel; // Canal de venda – IFOOD, EPADOCA, 99FOOD, ANOTAAI, DELIVERYVIP

scheduledAt: Date; // Para quando o pedido foi agendado (caso o orderTiming seja SCHEDULED)

status: EOrderStatus; // Estado do pedido – PENDING, PLACED, CONFIRMED, DISPATCHED, CONCLUDED

takeOutTimeInSeconds: number; // Tempo de retirada do pedido

takeout: IOrderTakeoutV3; // Objeto de informações do pedido em caso de retirada (TAKEOUT)

total: IOrderTotalV3; // Valores do pedido, total, entrega, descontos, adicionais

updatedAt: Date; // Data de atualização do pedido

version: string; // Versão do sistema em que o pedido foi feito

occurrences: CannoliIOrderOccurrence[]; // Lista das principais ocorrências ao decorrer do pedido

callbacks?: string[]; // Lista de callbacks para informativo externo

handshake: any; // Objeto informativo de disputas (somente pedidos originados do IFOOD), quando o cliente solicita um extorno, devolução, ou outros casos de negociação entre cliente e loja