# Arquitetura

COVER

# Modelo de domínio

O modelo de domínio é composto pelas entidades grupo, lista de compras, produto e assinatura, todas implementando a interface *Comparable* para poder comparar as entidades. Esta implementação serve para comparar o nome nas entidades grupo e lista e para comparar as assinaturas nos produtos, que comparam o tempo que foram registadas. Os tipos com ‘?’ indicam que o atributo pode ser NULL. O método *toMap* e o construtor *fromMap* servem para facilitar a interação com base de dados, que geralmente usam este tipo de estrutura de dados.

Diagram, schematic

Description automatically generated

# Realização de casos de utilização

## Caso de utilização – Comprar produtos

Como visto na especificação de requisitos, este caso de utilização começa por transitar para uma janela de compra após clicar no botão *BUY PRODUCTS*.

Ao transitar, a vistaModelo associada à próxima janela vai buscar o *snapshot* dos produtos que correspondem à lista em que o utilizador está. Este *snapshot* serve para receber um evento sempre que é alterada alguma coisa na base de dados, por exemplo, se outro utilizador adicionar um produto, vê-se o produto a ser adicionado no momento.

Depois, ao sinalizar os produtos como comprados/não-comprados, é atualizado o produto na base de dados com a assinatura de quem e quando foi comprado. Caso o utilizador retire a sinalização, esta assinatura desaparece.

Quando finalizadas as compras, o utilizador volta para a janela anterior clicando no botão para voltar atrás.

Diagram

Description automatically generated

Comprar produtos e criar grupo

# Arquitetura de mecanismos

# Arquitetura geral da solução