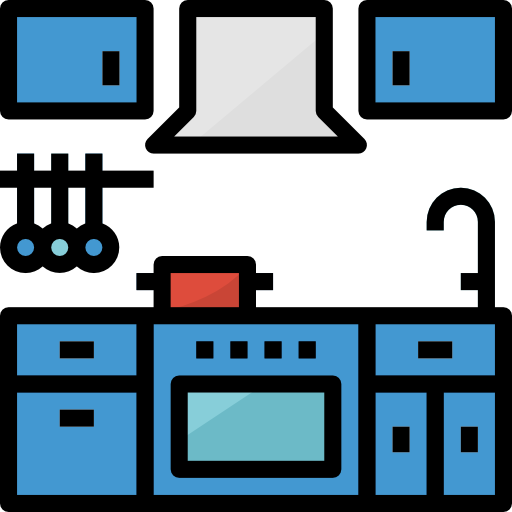


# 

# **Vamos ver outro exemplo**



Então, o que é uma dependência entre classes? É a necessidade de outro objeto para que uma classe funcione.

Se tivermos uma **classe Cozinha**, precisamos do **objeto Cozinheiro** para retornar os pedidos dos clientes.

**Atenção!** O objeto Cozinheiro é uma dependência da classe Cozinha.

Vamos pensar por um momento que devemos preparar o prato principal, então precisamos criar uma dependência do cozinheiro na nossa **cozinha**, porque é o **cozinheiro** que se encarrega de prepará-lo.

Ou seja, **criar uma instância do cozinheiro** dentro de nossa classe cozinha para passar, por exemplo, os ingredientes e o cardápio de que você precisa para preparar o prato principal.

Uma possível implementação disso poderia ser:

| public class CozinhaController {  private CozinheiroService umCozinheiro = new CozinheiroService();  /\*\* **@return** retorna o prato pronto\*/  public String prepararPrato(String ingredientes,String menu) {  umCozinheiro.setIngredientes(ingredientes);  umCozinheiro.setMenu(menu);  String pratoPronto = umCozinheiro.getPratoPronto();  return pratoPronto;  }  } |
| --- |

Criamos a instância CozinheiroService na classe CozinhaController, criando a dependência do cozinheiro com a cozinha.

O problema com essa abordagem é que, ao instanciar o objeto dentro da classe, estamos gerando um alto acoplamento. Isso torna nosso código não muito flexível, de modo que as alterações de uma classe pode afetar a outra. É por isso que devemos usar injeção de dependência e deixar que o framework se encarregue de injetá-la conforme a necessidade.

Mas… como as dependências são injetadas?