

1 Descrição

Um novo de modelo de negócio chegou a Joinville, o *Make Yourself Confeitaria*. Este sistema de franquias enlouquece os amantes de bolos e tortas por onde passa. Com foco nos desejos mais secretos do cliente, o modelo *Make Yourself* oferece aos clientes a personalização de suas sobremesas. O cliente monta seu próprio bolo ou torta. A confeitaria oferece um número de guichês (G) especializados em partes dos bolos e tortas. Para habilitar uma franquia, o proprietário deve escolher entre 3 e 5 guichês especializados.

Alguns exemplos de guichês sugeridos, representados por t_g , são: pandeló, coberturas, recheios, cortes, decoração. Os guichês são setores independentes coordenados pela recepção (*WS-Recepção*), que atua como intermediário. O cliente comunica-se somente com a *WS-Recepção*, que recebe as demandas e se responsabiliza pelo enfileiramento dos pedidos. De acordo com a política de atendimento definida, o *WS-Recepção* atribui um único pedido para cada guichê. Os guichês e a *WS-Recepção* desconhecem a receita dos clientes e a ordem de montagem da sobremesa.

A Figura 1 demonstra o esquema da Franquia de Joinville. O Sistema da Franquia de Joinville tem quatro guichês: *WS-Pandeló*, *WS-Cobertura*, *WS-Recheio* e *WS-Cortes*. (1) O guichê *WS-Pandeló* é responsável pela elaboração de

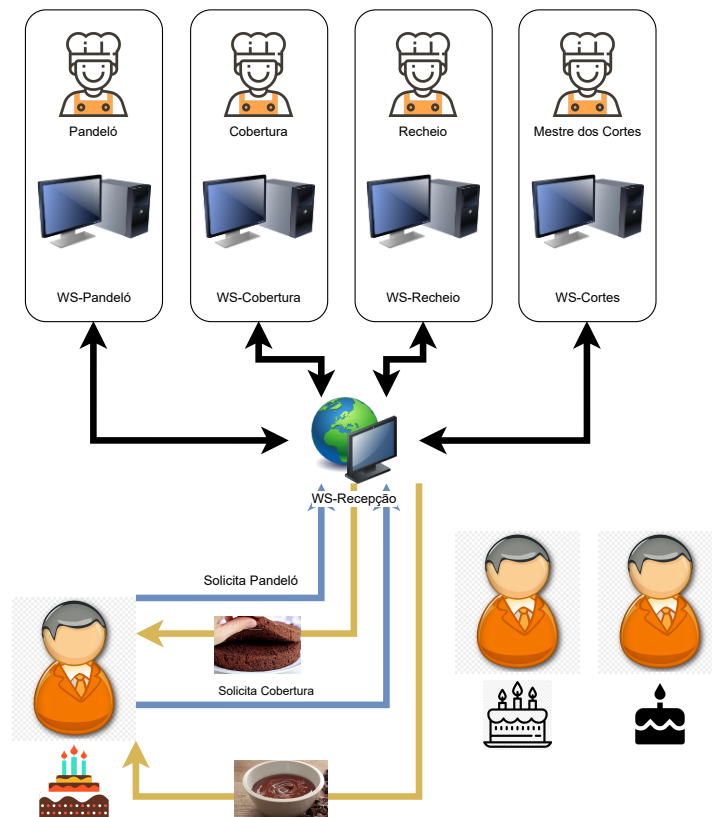


Figura 1: Funcionamento da Franquia do *Make Yourself Confeitaria*.

quatro tipos de pandeló: chocolate, baunilha, coco, cereais e milho. (2) O *WS-Cobertura* é mais limitado, oferece somente, duas opções: chocolate e calda de nozes. (3) Por outro lado, a equipe do recheio (*WS-Recheio*) caprichou no leque de opções: banana, morango, coco, leite condensado, suspiro, chantilly, chocolate amargo, chocolate ao leite, creme de cassis e trufas. (4) Finalmente, o Mestre dos Cortes (*WS-Cortes*) oferece serviços de corte como: bolo estrelar, torta montanha nevada, bolo circular, torta vulcão, bolo e torta traveseiro, e castelo. No esquema descrito acima, temos três clientes com três bolos diferentes. Um deles, o esquema especifica algumas transações. Pode-se observar que ele solicita o 'Pandeló', aguarda o término da solicitação e, em seguida, solicita a cobertura.

2 Arquivo de Entrada

A única forma de entrada de dados é por meio do arquivo de entrada, respeitando o formato pré-definido. Parte da informação do arquivo de entrada será destinada aos guichês e o restante aos clientes. Um exemplo de arquivo de entrada é o descrito na Figura 2. O *WS-Recepcao* lê a porção do arquivo que identifica os guichês disponíveis na Franquia, no caso de Joinville, os quatro descritos na figura, e entra no estado de espera de demandas. Após o lançamento dos guichês *WS-Pandelo*, *WS-Cobertura*, *WS-Recheio*, *WS-Cortes* e *WS-Recepcao*, o primeiro cliente é lançado. Neste exemplo, o Cliente 1 solicita, nesta ordem, os seguintes ingredientes: pandeló de baunilha, corte estrelar, recheios de trufas e banana, pandeló de chocolate, corte estrelar, recheios de chocolate ao leite e coco, finalizando, com a cobertura de chocolate. O Cliente 2 monta o seu bolo com pandeló de chocolate, corte circular, recheios de chocolate amargo e banana, pandeló de chocolate, corte circular, recheios de chocolate amargo e leite condensado, finalizando, com a cobertura de calda de nozes.

O arquivo de entrada determina a quantidade de guichês disponíveis (neste exemplo somente 4), de clientes (neste exemplo somente 2), e a lista de ingredientes solicitado por cada cliente. O lançamento do número total de guichês e clientes são especificados no arquivo de configuração (Seção 4).

```
## Recepcao ##
NClientes = 2
Guiche = WS-Pandelo
Guiche = WS-Cobertura
Guiche = WS-Recheio
Guiche = WS-Cortes

## Clientes ##
**** Cliente 1 ****
WS-Pandelo = baunilha
WS-Cortes = estrelar
WS-Recheio = trufas
WS-Recheio = banana
WS-Pandelo = chocolate
WS-Cortes = estrelar
WS-Recheio = chocolate ao leite
WS-Recheio = coco
WS-Cobertura = chocolate
**** Cliente 2 ****
WS-Pandelo = chocolate
WS-Cortes = circular
WS-Recheio = chocolate amargo
WS-Recheio = banana
WS-Pandelo = chocolate
WS-Cortes = circular
WS-Recheio = chocolate amargo
WS-Recheio = leite condensado
WS-Cobertura = calda de nozes
```

Figura 2: Exemplo de arquivo de entrada

3 Arquivo de Saída

O resultado final da aplicação é determinista. A análise do algoritmo se dará pela verificação do arquivo de saída (Figura 3). O método de comparação é o textual caractere a caractere, portanto é essencial que o formato especificado seja respeitado.

Os clientes (*NClientes*) devem ser lançados sequencialmente para garantir a ordem correta do texto de saída. No caso do exemplo, as seis primeiras linhas são geradas pelo *Cliente 1* e devem constar na saída padrão (stdout) do console do servidor. Em seguida, pode-se observar as seis linhas da saída do *Cliente 2*. Em ambos os casos, aparece o estado final do atendimento (*Pronto*) e quantidade de solicitações atendidas por cada guichê. Caso houvesse mais clientes, as demais linhas representariam os dados desses clientes.

Do lado dos fornecedores dos ingredientes, a mensagem final de contabilidade é de responsabilidade do *WS-Recepcao*. Ele gera o seguinte relatório: o estado final do atendimento (*finalizado*), o número de clientes atendidos (*Nro_clientes_atendidos*), o nome de cada guichê (*WS-Pandelo*, *WS-Cobertura*, *WS-Recheio*, *WS-Cortes* e *WS-Recepcao*) e a quantidade de ingredientes elaborados, respectivamente.

```
* SERVER: Beginning to publish WS Servers (ens1) *
* All done publishing. *
## Cliente (ens2) 1 ##
Status: Pronto
WS-Pandelo: 2
WS-Cobertura: 1
WS-Recheio: 4
WS-Cortes:2
#####
## Cliente (ens3) 2 ##
Status: Pronto
WS-Pandelo: 2
WS-Cobertura: 1
WS-Recheio: 4
WS-Cortes:2
#####
* Server End *
## Servidor ##
Status: finalizado
Nro_clientes_atendidos: 2
WS-Pandelo (ens1): 4
WS-Cobertura (ens1): 2
WS-Recheio (ens1): 8
WS-Cortes (ens1): 4
WS-Recepcao (ens1): 18
#####
```

Figura 3: Exemplo de arquivo de saída.

4 Tecnologia

O projeto da *Make Yourself Confeitaria* deve ser desenvolvido na linguagem de programação Java e se apoiar no *middleware* de comunicação *Web Service* (WS). No Git da disciplina é possível baixar um código exemplo de WS (<https://github.com/mauriciopillon/sdi0001/tree/master/05-WSJava-console>). As ferramentas de desenvolvimento estão instaladas na Plataforma de Desenvolvimento da disciplina (*Laboratório de Processamento Paralelo e Distribuído* (LabP2D)), e a aplicação deve, obrigatoriamente, ser validada na Plataforma de Submissão (BOCA) - (<http://boca-sdi.ddns.net/boca/src/>). O executável do servidor, **java MYC.ServerPublisher ens1**, deverá ser lançado na máquina *ens1*, via *script*, e os *N_{cli}* (**java Client ens1 1**, **java Client ens1 2**, ...) na máquina *ens2 ... ens4*. O arquivo *ambiente.in* (Figura 4) é um exemplo de configuração possível para execução. Apoiem-se nos *scripts* do problema teste, já disponível no BOCA, adequando o que for necessário.

```
Nro_clientes: 2
Processos: Client Client
Maquinas: ens2 ens3
```

Figura 4: Exemplo do arquivo de configuração.