

Pré-Requisitos

O projeto terá que ser desenvolvido na linguagem de programação C e usando as técnicas adequadas lecionadas nas aulas teórico-práticas e laboratoriais.

O IDE a utilizar fica ao critério dos alunos mas, caso não utilizem o IDE usado nas aulas (NetBeans), terão que (antes de submeterem) criar os respetivos projetos finais no IDE NetBeans.

Objetivo Principal

Compreensão das estruturas FIFO (First In First Out) e LIFO (Last In First Out) através da implementação estática ou dinâmica dos TADs (Tipos Abstratos de Dados) filas e pilhas.

Descrição do Problema

O objetivo deste projeto é implementar uma pequena aplicação para gestão de clientes com cartões de fidelização para um restaurante de sushi onde os pratos circulam num tapete rolante (como, por exemplo, no Yo!Sushi¹). Os pratos têm cores que correspondem a diferentes preços (ver Figura 1).

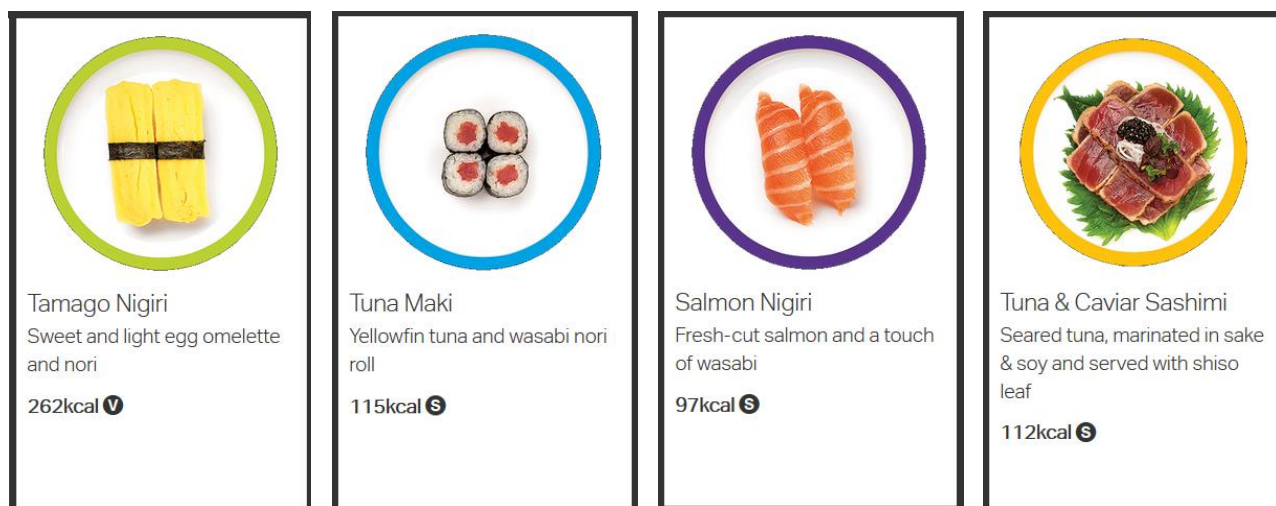


Figura 1- Exemplos de pratos com rebordos de cores diferentes indicando o preço da peça.

Os clientes retiram cada prato pretendido do tapete (também o podem pedir ao funcionário) e, após consumo da respetiva peça, empilham-no resultando numa **pilha de pratos** consumidos ao

¹ <https://yosushi.com/food>

longo **da refeição** (ver Figura 2). No final da mesma, o funcionário do restaurante vai **desempilhar os pratos para obter o valor final da refeição**.



Figura 2 – Pratos empilhados após a refeição.

A criação do cartão de **cliente** faz-se por **registo** deste último, processo esse que passa pela introdução do seu **nome**, **apelido**, e **idade** (parte-se do princípio que a idade não se altera ao longo do tempo por questões de simplificação do problema e visto não ser essencial para o objetivo do projeto). O **cartão** associado a um cliente tem também o **número** (único) que o identifica e o registo das **refeições consumidas pelo cliente no último mês**.

Os **dados** de uma **refeição** ficam **guardados** apenas durante **um mês**. Por exemplo, ao dia 15 de Maio, o cartão terá o registo de todas as refeições efetuadas a partir de 16 de Abril. Sempre que os dados de uma refeição são eliminados, o **preço final da refeição é convertido** diretamente em **pontos** (inteiro) a acumular no seu cartão, que assim vai acumulando os pontos obtidos para todas as refeições que já não se encontram registadas, correspondendo a uma espécie de histórico que não inclui o mês mais recente (últimos 30 dias).

Uma **refeição** é registada (**iniciada**) com a **data atual** da aplicação (simulada através de uma operação) e **sem pratos** associados, que depois vão sendo “consumidos” (empilhados) ao longo da mesma. Parte-se do princípio que um prato retirado do tapete é empilhado, mesmo que ainda não tenha sido “consumido”. O **cliente** pode recolocar no tapete (**devolver**) o **último prato** que retirou do mesmo. O cliente **não pode repetir um prato** na mesma refeição.

No final da refeição, perdem-se os dados específicos dos pratos “consumidos”. Assim sendo, apenas se mantém o registo dos dados dos pratos enquanto a refeição estiver “ativa”. Quando a refeição é concluída, os seus **pratos** devem ser **contabilizados** para se obter o **preço total final**. O cliente pode **usar 100 pontos** (abater dos pontos acumulados no cartão) para ter um desconto de 10€ na conta final, desde que tenha efetuado **3 refeições nos últimos 30 dias**; ou pode abater **200 pontos** para obter um desconto de 20€, se tiver efetuado **5 refeições** nos 30 dias antecedentes.

Finalmente, é importante referir que o restaurante é muito popular, conceituado e procurado, onde os pratos presentes no tapete correspondem a um menu de degustação preparado pelo chefe para cada noite (apenas serve jantares, fechando às segundas, terças e quartas). O restaurante não aceita reservas (marcações por hora), apenas apresentando **4 lugares** ao balcão, tendo que existir uma **fila de espera** com base na ordem de chegada dos clientes (que apresentam o cartão, obrigatório). Contudo, é dada **prioridade** a clientes com **idade superior a 60 anos**. Quando um lugar ao **balcão é libertado** (fatura emitida), o **primeiro da fila** de espera ocupa esse lugar.

Desenvolvimento do Trabalho

Para o desenvolvimento do trabalho será necessário:

- implementar uma aplicação (os alunos escolhem se baseada numa implementação estática ou dinâmica dos TADs envolvidos) de consola (interfaces visuais não são consideradas para a avaliação) que resolvam o problema descrito acima;
- e escrever um pequeno relatório.

O código deverá ser escrito tendo em conta as regras de boa programação e de acordo com as convenções seguidas na disciplina.

É da responsabilidade dos alunos decidir quais os TADs e respetiva implementação que melhor se adequam à informação a armazenar. Os projetos devem usar tanto pilhas como filas.

Não é permitido alterar a estrutura dos ficheiros nem acrescentar informação adicional, como por exemplo, caracteres de terminação.

Casos omissos ou inconsistentes na especificação deverão ser tratados da forma que se achar mais correta, sob orientação do respetivo docente de laboratório.

Informação a armazenar

1. Data:
 - dia (inteiro)
 - mês (inteiro)
 - ano (inteiro)
2. **Cartões de clientes**, em que cada cartão guarda a seguinte informação:
 - número de cartão (inteiro)
 - apelido (cadeira de caracteres)
 - nome próprio (cadeira de caracteres)
 - idade (inteiro)
 - sequência recente de **Refeições**
 - pontos acumulados (inteiro)
3. Refeição:
 - **data da refeição** (Data)
 - **Pratos** empilhados
 - valor final (real)
4. **Pratos disponíveis**, em que cada prato tem a seguinte informação:
 - Número de prato (inteiro)
 - nome da peça (cadeira de caracteres)
 - cor do prato (cadeira de caracteres)
 - preço (real)

Dados de Entrada

A aplicação receberá os dados de entrada através dos seguintes ficheiros de texto:

- **"clientes.txt"**, que contém os dados de cada cliente por linha, com o seguinte formato:
 <numeroCartao> <apelido> <nomeProprio> <idade>
 Exemplo:
 10 Oliveira Rute 25
- **"pratos.txt"**, que contém os dados de cada prato por linha, com o seguinte formato:
 <numeroPrato> <nomePeca> <preco> <corPrato>
 Exemplo:
 1 TunaMaki 3.0 azul
- **"operacoes.txt"**, que contém uma sequência de operações para uso do sistema, visando testar o funcionamento do mesmo. Cada linha corresponde a uma operação, que será

assumidamente reconhecida, não necessitando de ser validada, mas tendo que produzir um resultado válido.

Operações possíveis:

- **AD <data>: Atualizar Data** na aplicação, com indicação da data no formato "dd/mm/aaaa", que tem o objetivo de simular a passagem de tempo (de datas). Parte-se do princípio que a operação é sempre usada com uma data mais atual que na execução anterior.

Output:

- Nova data: <dd/mm/aaaa>

- **AC <numeroCartao>: Apresenta-se o Cliente** no restaurante, com indicação do número de cartão.

Outputs possíveis:

Caso a operação não tenha sucesso

- Cliente desconhecido

Caso a operação tenha sucesso

- Cliente colocado ao balcão
- Cliente em fila de espera na posição <posição>

- **IR <numeroCartao> <data>: Iniciar uma Refeição**, com indicação do número de cartão e data no formato "dd/mm/aaaa". Em alternativa à indicação da data, pode-se registar na refeição a data que se encontre no sistema.

Outputs possíveis:

Caso a operação não tenha sucesso

- Cliente desconhecido
- Cliente não está ao balcão

Caso a operação tenha sucesso

- Refeição de cliente <nome> <apelido> iniciada a <dd/mm/aaaa>

- **FR <numeroCartao> <pontos>: Finalizar uma Refeição**, com indicação do número de cartão. Indica-se também o número de pontos que se pretende descontar (0, 100 ou 200). Apresenta a listagem de pratos consumidos e a conta final. Parte-se do princípio que é referente a refeição iniciada, caso cliente esteja ao balcão. Liberta lugar ao balcão.

Outputs possíveis:

Caso a operação não tenha sucesso

- Cliente desconhecido
- Cliente não está ao balcão
- Número de pontos inválido

Caso a operação tenha sucesso (exemplo para 3 pratos)

- Prato <numeroPrato> (<corPrato>): <nomePeca> <preco>€
Prato <numeroPrato> (<corPrato>): <nomePeca> <preco>€
Prato <numeroPrato> (<corPrato>): <nomePeca> <preco>€
Desconto aplicado: <desconto>€
Valor a pagar: <valorFinal>€

- **RP <numeroCartao> <numeroPrato>: Recolhe Prato**, com indicação do número de cartão e do número de prato. Cliente recolhe o prato para consumo a partir do tapete ou entregue pelo funcionário. Parte-se do princípio que é referente a refeição iniciada, caso cliente esteja ao balcão.

Outputs possíveis:

Caso a operação não tenha sucesso

- Cliente desconhecido
- Cliente não está ao balcão
- Prato desconhecido

Caso a operação tenha sucesso

- Empilhado o prato <numeroPrato> (<nomePeça>)

- **DP <numeroCartao>**: Devolve **Prato**, com indicação do número de cartão. Cliente recoloca no tapete o prato mais recente. Parte-se do princípio que é referente a refeição iniciada, caso cliente esteja ao balcão.

Outputs possíveis:**Caso a operação não tenha sucesso**

- Cliente desconhecido
- Cliente não está ao balcão

Caso a operação tenha sucesso

- Devolvido ao tapete o prato <numeroPrato> (<nomePeça>)

- **UP <numeroCartao>**: Último **Prato** consumido, com indicação do número de cartão. Parte-se do princípio que é referente a refeição iniciada, caso cliente esteja ao balcão.

Outputs possíveis:**Caso a operação não tenha sucesso**

- Cliente desconhecido
- Cliente não está ao balcão

Caso a operação tenha sucesso

- Último consumo de prato <numeroPrato> (<nomePeça>)

- **CP <numeroCartao>**: Conta **Pratos** consumidos até ao momento na refeição, com indicação do número de cartão. Parte-se do princípio que é referente a refeição iniciada, caso cliente esteja ao balcão.

Outputs possíveis:**Caso a operação não tenha sucesso**

- Cliente desconhecido
- Cliente não está ao balcão

Caso a operação tenha sucesso

- Contagem de pratos: <contaPratos>

- **VC <numeroCartao>**: Visualizar dados de **Cartão**, com indicação do número de cartão, tendo o objetivo de consultar o total de pontos e o número de refeições efetuadas no último mês.

Outputs possíveis:**Caso a operação não tenha sucesso**

- Cliente desconhecido

Caso a operação tenha sucesso

- Cliente <nomeProprio> <apelido> com <totalPontos> pontos acumulados e <numeroRefeicoes> refeições no último mês

- **CE**: listagem de clientes em espera por lugar ao balcão, apresentando também a contagem.

Outputs possíveis:**Caso a operação não tenha sucesso**

- Fila de espera inexistente

Caso a operação tenha sucesso (exemplo de 2 clientes)

- <contagemClientesEmFila> clientes em fila de espera
<apelido> <nomeProprio> (<idade> anos)
<apelido> <nomeProprio> (<idade> anos)

Os resultados obtidos com as operações submetidas através do ficheiro "operações.txt" deverão ser armazenados num ficheiro de texto chamado "resultados.txt".

Apesar do output pretendido ser gravado num ficheiro, pretende-se a escrita dos resultados também no ecrã com informação relevante às operações que estão a ser realizadas nas estruturas de dados e qual a informação gerada. Esta informação é útil durante o desenvolvimento do projeto e pode ser útil no momento da discussão em benefício dos alunos dado que permite uma melhor compreensão do que foi implementado.

Para efeitos de testes, a seu tempo serão disponibilizados um ficheiro "operações.txt" e o correspondente "resultados.txt".

Critérios de Avaliação

É obrigatório que os dados de entrada sejam lidos a partir de ficheiros e que os resultados de saída sejam escritos num ficheiro.

A avaliação do trabalho será feita de acordo com os seguintes princípios:

- Estruturação: o programa deve estar estruturado em TADs considerados necessários e convenientes para a resolução do problema.
- Correção: o programa deve executar todas as funcionalidades, tal como pedido.
- Legibilidade e documentação: o código deve ser escrito, formatado e comentado de acordo com o standard de programação definido para a disciplina.

Implementações que não sigam estes princípios poderão ter uma forte penalização.

As cotações são organizadas segundo a tabela seguinte.

Descrição	Cotação
Implementação dos TAD	4 valores
Leitura e tratamento dos dados de clientes e de pratos	3 valores
Leitura e tratamento das operações	1,5 valores
Implementação dos algoritmos/operações referentes à fila de espera	2,5 valores
Implementação dos algoritmos/operações referentes à pilha de pratos	2 valores
Implementação dos algoritmos/operações referentes às refeições	1 valor
Atualização da data e da sequência de refeições com base na data	2 valores
Escrita dos resultados em ficheiro	1 valor
Implementação no geral	1 valor
Relatório	2 valores

Relatório

No relatório deverão constar as seguintes secções:

- a) Descrição dos tipos abstratos de dados usados e respectivas implementações.
- b) Descrição dos principais algoritmos desenvolvidos.
- c) Conclusões e Limitações.

Instruções e Regras Finais

O não cumprimento das regras a seguir descritas implica uma penalização na nota do trabalho prático. Se ocorrer alguma situação não prevista nas regras a seguir expostas, essa ocorrência deverá ser comunicada ao respetivo docente de laboratório de ATAD.

Regras:

- a) O Mini-Projeto deverá ser elaborado por **dois alunos do mesmo docente**.
- b) A nota do Mini-Projeto será atribuída individualmente a cada um dos elementos do grupo após a discussão. As discussões poderão ser orais e/ou com perguntas escritas. As orais poderão ser feitas com todos os elementos do grupo presentes em simultâneo ou individualmente.
- c) A apresentação de relatórios ou implementações plagiadas leva à imediata atribuição de nota zero a todos os trabalhos com semelhanças, quer tenham sido o original ou a cópia.
- d) No rosto do relatório e nos ficheiros de implementação deverá constar o número, nome e turma dos autores e o nome do docente a que se destina.
- e) O trabalho deverá ser submetido no moodle, no link do respetivo docente de laboratórios criado para o efeito, até às 23h59 do dia 05 de Maio de 2017. Para tal terão que criar uma pasta com o nome: nomeAluno1_númeroAluno1-nomeAluno2_númeroAluno2, onde colocarão o ficheiro do relatório em formato pdf e as duas pastas com os projetos NetBeans da implementação das aplicações a desenvolver. Os alunos terão de submeter essa pasta compactada. Apenas será permitido submeter um ficheiro.
- f) Não serão aceites trabalhos entregues que não cumpram na íntegra o ponto anterior.
- g) As datas das discussões serão publicadas após a entrega dos trabalhos.