

Disciplina	LG1A1	Ano/Semestre Letivo	2020/01
Professores	Eurides Balbino/André Evandro	Tipo avaliação	Projeto Semestral
Aluno		Prontuário	

PROJETO SEMESTRAL

Objetivo:

Gerar a aplicação (em Linguagem C) para gerenciar o drive thru do Mickey & Donald

O sistema tem por objetivo realizar a venda de produtos alimentícios, gerenciando a oferta (menu de produtos), o pedido do cliente, a forma de pagamento e a fila de atendimentos.

Sobre o funcionamento da aplicação

O sistema deve oferecer os módulos:

Módulo de Vendas (“Faça seu pedido”) - onde constarão os itens armazenados numa base de dados chamada PRODUTOS.DAT.

Haverá uma estrutura na qual devem constar:

Codprod	Inteiro	Código do produto - chave primária (autonumérico)
Nomeprod	Texto	Nome do produto
Custoprod	Moeda	Valor unitário do produto

Neste módulo você deverá compor o pedido do cliente da seguinte maneira:

O cliente escolhe qual produto e quantas unidades deseja comprar e o sistema compõe o pedido registrando o código do pedido com os itens do pedido - código, nome do produto, custo unitário, quantidade e valor do item pedido.

Ao final do pedido deve constar o valor total da compra.

Exemplificando:

Menu de produtos

Codprod	Nomeprod	Custoprod
1	Lanche número 1	R\$ 9,50
2	Lanche número 2	R\$ 8,50
3	Lanche número 3	R\$ 7,50

Pedido Nro = 001

Itens do pedido

Código	Produto	Custo unitário R\$	Quantidade pedida	Valor do item pedido R\$
2	Lanche número 2	8,50	2	17,00
1	Lanche número 1	9,50	2	19,00
Total da compra				36,00

Módulo de Cobrança (“Pague”) - neste módulo serão oferecidas as seguintes formas de pagamento: dinheiro, cartão de crédito/débito ou cheque.

A partir do momento em que o cliente opta por uma forma de pagamento, o aplicativo deverá registrar na base de dados PAGAMENTOS.DAT os dados a seguir.

Campos da estrutura de dados de PAGAMENTOS.DAT:

codpgto (autonumérico) = código do pedido

formapgto (vetor de caracteres) = dinheiro/cartão de crédito/cartão de débito/cheque

valorpcto (float) = valor total do pedido

No caso do pagamento com cartão de crédito/débito, o aplicativo deverá validar o cartão da seguinte maneira: o número do cartão deverá conter exatamente 16 dígitos; em caso contrário emitir a mensagem “cartão inválido” e solicitar o número do cartão novamente.

Uma vez aceito o cartão de crédito/débito, o sistema deverá registrar na base de dados CARTOES.DAT o código do pedido pago com cartão de crédito/débito e o número do cartão que deverá ser “mascarado”, ou seja, os dígitos de número 5 a 12 deverão ser trocados por asteriscos (*) quando o cartão for gravado em CARTOES.DAT.

Campos da estrutura de dados de CARTOES.DAT:

Codpgto (número) = código do pedido que foi pago com cartão de crédito/débito.

Numcartao (vetor de caracteres) = número do cartão de crédito/débito.

Exemplificando: o pedido nro 3 foi pago com cartão cujo número é 1111222233334444. Em CARTOES.DAT, foram registradas as informações conforme ilustrado abaixo.

PAGAMENTOS.DAT			CARTOES.DAT	
codpgto	formapgto	valorpcto	codpgto	numcartao
1	dinheiro	R\$ 19,00	3	1111*****4444
2	cheque	R\$ 8,50	0	
3	cartão	R\$ 25,50		

Módulo de Espera (“Fila”) - neste módulo é gerenciada a fila de espera de clientes. Assim, deve-se assumir que “o primeiro que entra é o primeiro que sai”. Aqui o aplicativo deve prever a situação em que o cliente deve permanecer na fila de espera. Tal situação ocorre quando, ao tentar atender um cliente é detectado que seu pedido vai demorar a ser completado e, para não atrasar os demais, o cliente é encaminhado para a fila de espera, na qual também é obedecido o critério de: “o primeiro que entra é o primeiro que sai”.

Módulo de Entrega (“Leve”) - neste módulo os produtos adquiridos pelo cliente são entregues somente depois de efetuado o pagamento do pedido.

Assim, depois de realizada a cobrança, a aplicação deverá exibir em tela a seguinte informação: PEDIDO PAGO COM XXXXXXXX (onde XXXXXXXX significa dinheiro ou cheque ou cartão de crédito/débito) nas cores: verde para dinheiro; azul para cheque e lilás para cartão de crédito/débito.



