

## 1 Grupo 1: Controle de figurinhas copa

O programa será utilizado para realizar o controle de figurinhas da copa. O programa permitirá o cadastro de novas figurinhas no banco de dados, verificando se a mesma já existe ou não, bem como a extração de indicadores sobre a coleção. As seguintes funcionalidades devem estar implementadas:

1. Uma tela para o cadastro de novas figurinhas, com os campos de busca de número de figurinha e país associado. O cadastro só é executado se a figurinha ainda não existe, se ela já existe, uma mensagem deve avisar ao usuário.
2. Um botão deve gerar um relatório atualizado com as seguintes informações:
  - (a) Número de figurinhas faltantes para cada país.
  - (b) Porcentagem faltante de figurinhas para cada país.
  - (c) Qual é o país crítico no momento (que precisa de mais figurinhas para se completar o álbum).
  - (d) % total do álbum completa.
3. Um outro botão deve limpar os dados gerados no relatório.

## 2 Grupo 2: Jogo do labirinto

O programa implementa um jogo do labirinto, em que o usuário deve direcionar um objeto (personagem), até que ele consiga sair do labirinto de obstáculos. As seguintes funcionalidades devem ser implementadas:

1. O mapa, ou seja, a posição dos obstáculos deve ser reconfigurável, ou seja, deve existir algum arquivo (que pode estar na própria planilha), em que podemos determinar como o mapa será composto, de forma que quando o jogo é inicializado este arquivo é 'transcrito' em um desenho na planilha.
2. O mapa deve ser fechado nas bordas, exceto por uma saída. O jogo pode terminar se uma de duas situações ocorrer: se o personagem 'bater' em um obstáculo (o usuário perde o jogo), ou se o personagem sai do labirinto.
3. Toda vez que houver uma colisão entre personagem e obstáculo o jogo deve parar com a mensagem de fim de jogo.

## 3 Grupo 3: Controle de estoque

O programa deve realizar o controle de estoque de uma empresa, bem como gerenciar o estoque mínimo do produto. As seguintes funcionalidades devem ser implementadas

1. Cadastro de produtos: deve existir um menu para se inserir novos produtos (nome, descrição, quantidade em estoque, estoque mínimo) que serão salvos em um banco de dados. Caso o produto já exista, o sistema deve avisar o usuário.
2. Alteração de produtos: deve existir um campo para 'busca' de produtos, em que (se existir), todas as informações do produto são mostradas, e o usuário pode realizar alterações nas informações, que serão também atualizadas no banco de dados.
3. Entrada/saída de produtos do estoque: O usuário deve ser capaz de adicionar ou remover quantidades de produto, de forma que o estoque seja atualizado automaticamente. Na opção de retirada de estoque, dois casos podem ocorrer:
  - (a) Quantidade restante após a operação  $< 0$ : Nesse caso a operação é cancelada, e o usuário é avisado de que não existe estoque suficiente para a remoção.
  - (b) Quantidade restante após a operação  $<$  estoque de segurança: Um alerta deve ser emitido ao usuário para que ele reponha o estoque.

## 4 Grupo 4: Jogo da minhoca simplificado

O jogo permite ao usuário controlar um personagem (objeto), em um mapa com obstáculos. O jogador deve controlar o objeto até que ele atinja um objeto alvo, sem colidir com os obstáculos. As seguintes funcionalidades devem ser implementadas:

1. O mapa, ou seja, a posição dos obstáculos deve ser reconfigurável, ou seja, deve existir algum arquivo (que pode estar na própria planilha), em que podemos determinar como o mapa será composto, de forma que quando o jogo é inicializado este arquivo é 'transcrito' em um desenho na planilha.
2. O mapa deve conter um objeto alvo, que é desenhado de forma a se destacar dos outros.
3. Toda vez que houver uma colisão entre personagem e obstáculo o jogo deve parar com a mensagem de fim de jogo.
4. Quando o usuário chegar ao valor alvo, sem atingir os obstáculos a mensagem de "vitória" deve aparecer na tela.

## 5 Grupo 5: Jogo de perguntas

O programa deve ter um formulário (para os gerenciadores preencherem), com novas perguntas que podem ser adicionadas a um banco de dados de perguntas. O jogo consiste em realizar perguntas ao usuário, que deve respondê-las. Por fim, o seu acerto total é calculado e mostrado. As seguintes funcionalidades devem ser implementadas:

1. Menu para o cadastro de novas perguntas em um banco de dados.
2. Quando o jogo começar, o número de perguntas deve ser definido. Com esse número as perguntas devem ser feitas de forma aleatória para o usuário, sem repetição.
3. Ao fim de todas as perguntas o percentual de acertos do usuário deve ser calculado.

## 6 Grupo 6: Controle de certificados

O programa vai controlar a data de expiração de dois tipos de licenças que os funcionários possuem, bem como servir de alerta para as datas mais críticas. O programa deve ter as seguintes funcionalidades implementadas:

1. Menu para o cadastro de novos funcionários, bem como seus certificados, e datas de vencimento. Se o cliente já existe, então o sistema deve avisar o usuário, e não realizar a inserção.
2. Menu para alteração de dados de funcionários. Caso um funcionário renove o certificado, por exemplo.
3. Relatório de controle: um botão deve ser inserido para gerar um relatório sobre os funcionários com os seguintes dados:
  - (a) Porcentagem de funcionários com e sem os certificados.
  - (b) Dos funcionários que têm o certificado, porcentagem que está com o mesmo vencido e porcentagem "oK".
  - (c) Funcionário crítico: dos funcionários que estão com o certificado vencido, qual é o que está vencido há mais tempo.

## 7 Grupo 7: Calculo do BOM (*Bill of Materials*)

O programa deve "ler" uma árvore de produtos, e receber do usuário um produto e uma quantidade que deve ser produzida. Com isso, calcular a lista de todas as quantidades de materiais na árvore que serão necessários para produzir o produto "pai". Por exemplo:

O cálculo do BOM está associado a explosão dos componentes de um produto em forma de árvore. Seja a árvore de produto seguir (1), que demonstra as relações de precedência entre todos os componentes (dentro dos círculos o número do item e ao lado a quantidade necessária).

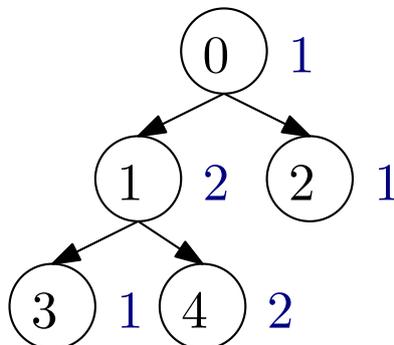


Figura 1: Árvore de produto

O problema de calcular o BOM é: Dada uma demanda do produto 0, quanto deverá ser produzido (ou comprado) de todos os elementos de que ele é composto, para a produção de 0? No caso da árvore da Figura 1, para uma demanda de 2 unidades de 0, são necessárias as seguintes quantidades de 1,2,3 e 4:

- 1: 4
- 2: 2
- 3: 4
- 4: 8

## 8 Grupo 8: Controle de CRM

O programa vai fazer um controle de CRM em relação às últimas compras dos usuários. Um relatório pode ser gerado diariamente que informa para o usuário quais clientes precisam de um atendimento no pós venda (considerando um período  $n$  após a última compra do cliente). O programa deve ter as seguintes funcionalidades, baseado na Figura 2.

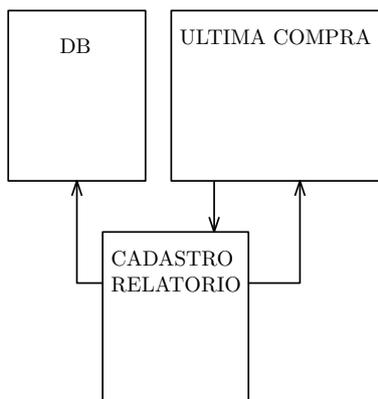


Figura 2: Dinâmica CRM

1. Uma aba para o menu de cadastro da compra dos clientes (CADASTRO RELATÓRIO), com um botão para a inserção. Quando o botão é clicado, os dados da compra do cliente devem ser inseridos em um banco de dados de transações (DB).
2. Ao se inserir uma nova transação, a planilha ULTIMA COMPRA também deve ser atualizada. Se essa for a primeira compra do cliente, então um novo registro deve ser feito. Caso contrário, o campo referente a última data de compra do cliente deve ser atualizado com a data da nova compra.
3. Na aba CADASTRO RELATORIO, também deve existir um botão de gerar relatório. Esse botão deve verificar se existe algum cliente que deve ser contactado hoje, ou seja, se a última compra dele + um  $\Delta T$  é igual ao dia de hoje, e mostrar os clientes na planilha.

## Referências