Gestão de Piscinas

Mestrado Integrado em Engenharia Informática e Computação

Algoritmos e Estruturas de dados

Turma 1

Elementos do grupo:

Cristiana Maria Monteiro Ribeiro - 201305188 - up201305188@fe.up.pt

Mário Gustavo Gomes Rosas de Azevedo Fernandes - 201201705 – up201201705@fe.up.pt

Sérgio António Dias Salgado - 201406136 - up201406136@fe.up.pt

2017/01/01

Índice

Descrição sucinta do tema

Esta aplicação visa facilitar a gestão de uma piscina, de modo a garantir a prestação de serviços de qualidade aos seus utentes. Para isso, é útil fazer o controlo de pessoal, quer seja dos utentes, quer seja dos professores. O sistema é capaz de registar todas as ocorrências em que um utente utiliza a piscina.

Uma piscina aufere de diversos horários de funcionamento, sendo necessário defini-los para a utilização organizada por parte dos utentes. Estes podem optar por frequentar a piscina em dois modos de aula distintos: aula livre e aula acompanhada.

Cada aula tem um custo associado diferente e pretende-se que esse custo seja apresentado ao utente em forma de fatura mensal. Aos utentes é permitido fazer depósitos na sua conta, e assim sendo, torna-se possível debitar automaticamente o valor das suas despesas. No caso do saldo ser insuficiente, o utente é notificado com uma mensagem a fim de regularizar as suas contas.

Para além de um custo fixo, cada tipo de aula respeita uma duração pré-estabelecida. Uma aula acompanhada tem a duração de uma hora, enquanto que uma aula livre pode durar até duas horas (divididas em períodos múltiplos de trinta minutos).

Uma aula acompanhada tem um professor que é atribuído através da nossa aplicação, onde se garante que os professores são distríbuidos de forma a que o número de aulas por professor seja equivalente para todos os docentes, dependendo dos seus horários. Os professores são registados no sistema de modo a que seja possível adicionar, editar e remover um professor, bem como obter a sua listagem.

O mesmo se passa com os utentes, sendo que todas as operações descritas anteriormente são válidas para um utente. O sistema também é capaz de associar os utentes às aulas que pretendem frequentar. Para facilitar o processo, e tendo em conta que as aulas acompanhadas têm prioridade sobre as aulas lives, o utente pode escolher o horário que mais lhe convém. Se o utente pretende frequentar uma aula acompanhada e o número máximo de ocupantes da piscina tiver sido atingido, o programa remove o último incrito na aula livre a decorrer em simultâneo. No caso de não existir, ou seja, a aula acompanhada está com o número máximo de utentes que podem estar a frequentar a piscina, o sistema simplesmente ignora o pedido, comunicando que a operação não pode ser feita.

Neste sistema pode proceder-se à consulta de informação relevante, portanto, é possível saber os utentes e professores inscritos na piscina, as aulas que estes

frequentam e o respetivo horário. Também é fácil obter listagens detalhadas das aulas e dos horários existentes, bem como dos utentes e dos professores.

Salienta-se a capacidade que o sistema tem para atualizar, diariamente, os dados relativos às aulas que o utente já frequentou até à data, bem como a visualização de estatísticas num determinado período de tempo. Por último o programa também fornece a capacidade de prever para um dado tempo quantas aulas vao ocorrer, quanto dinheiro irá ser gerado (tendo em conta o numero de utentes em cada aula) e ainda a quantidade de utentes que vão estar presentes no total dessas aulas.

Os dados alterados ao longo da execução da aplicação são guardados para utilizações futuras nos ficheiros respetivos aquando da terminação do programa.

Para além das funcionalidades enunciadas, com a atualização da aplicação, a piscina possui agora uma loja onde é possível comprar produtos, editá-los, listá-los e abastecer a loja.

Com a adição de coordenadas na informação da piscina, podemos localizar a piscina mais próxima em que determinada aula é leccionada, visualizar todas as piscinas que leccionam uma determinada aula e ainda listar todas as piscinas ordenadas em função da distância a que se encontram da piscina que está a ser gerida pela aplicação.

Se um utente não frequentar a piscina por um período superior a trinta dias, esse utente é marcado como inativo, e para os utentes em estado inativo é feita uma campanha para que estes atualizem as suas informações no caso de serem inválidas. Podemos ainda listar todos os utentes inativos ou remover todos os utentes inativos da tabela de inatividade, passando estes a estar novamente ativos.

Descrição da solução implementada

Depois de uma análise sucinta às funcionalidades do sistema, começamos por descrever o modo como foram desenvolvidas.

Foi criado um mecanismo de segurança para que seja possível aceder à aplicação. Primeiro é validado o nome da piscina, e se este existir é pedido ao utilizador que introduza a respectiva *password*. Se esta estiver errada, o programa termina. Caso contrário, se os ficheiros relativos a essa piscina não estiverem corrompidos, são carregados no sistema e os dados alterados em execuções anteriores são preservados.

Se o sistema não contiver o nome da piscina, é possível criar uma nova e para isso é suficiente introduzir o nome da piscina e a nova *password*.

A interface do sistema é composta por vários menus (utentes, professores, aulas, horários) onde estão listadas as várias operações que é possível realizar. As ações de *create, read, update* e *delete* estão presentes em todos eles. Foi feita a validação dos *inputs* do utilizador na seleção das várias opções.

Como foi dito anteriormente, é obrigatório o tratamento das pessoas que frequntam a piscina de forma organizada. Desta forma, criou-se a class Pessoa que tem como atributos o id, o nome e a idade. As classes derivadas, Utente e Professor, para além desses atributos têm o membro-dado saldo e um vetor do tipo Horario que é responsável por guardar os horários da disponibilidade de um professor.

Relativamente à gestão das aulas, foi criada a classe Aula que tem como dados membro a identificação da aula, o horário da aula e um vector dos utentes que participam nesta aula. A classe Aula é herdada pelas classes Aula Livre e Aula Acompanha que identificam cada tipo aula. A aula acompanhada tem como atributos o preço da aula e o professor que vai leccionar esta aula. A class Aula Livre tem como atributos o preço da aula.

Para ser possível definir os diferentes horários de cada aula, bem como o horário de funcionamento da piscina, foi desenvolvida a class Horario, que tem como dados membro o dia da semana, uma data inicial e uma data final do tipo Data. A class Data tem como atributos o dia, o mês, o ano, a hora e os minutos que interessam para estruturar uma data.

A fim de auxiliar o desenvolvimento deste projeto, foram criadas os ficheiros .cpp Menu, Auxiliary, Interacao e MyExceptions. Auxiliary contém funções úteis ao programa, entre as quais, funções de verificação de condições requeridas para a compatibilidade com a estrutura da aplicação.

Como o nome indica, Interacao implementa as funções que requerem interação com o utilizador da aplicação, sendo que Menu irá tratar da interação com os menus. Neste ficheiro encontram-se as funções para criar, editar, listar e remover objetos das classes apresentadas anteriormente (Utente, Professor, Aula e Horario) e, mediante os *inputs* do utilizador, chamam as devidas funções na classe Piscina. Também está implementada a função predictNext(), cujo papel é o cálculo de estatísticas relacionadas com a piscina, de acordo com um determinado espaço de tempo escolhido pelo

utilizador. É possível visualizar o número de aulas planeadas, a receita gerada por essas aulas e o número de utentes inscritos nessas aulas.

De forma a garantir que estes são válidos e não comprometem a execução da aplicação, foi criada a classe MyExceptions para o tratamento das várias exceções que possam surgir.

Para concluir foi desenvolvida a class Piscina, cujos atributos são: o nome da piscina, os nomes dos ficheiros de utentes, professores, horários e aula (relativos à piscina instanciada), e vetores do tipo Utente, Professor, Horario e Aula onde é guardada a informação. Existem dois vetores do tipo Aula que são: aulas e aulas_dadas, que diferem no seguinte: o vetor aulas contém as aulas que ainda vão ser leccionadas e o vetor aulas_dadas contém as aulas que já foram leccionadas até à data atual. O vetor aulas_dadas é preenchido aquando da execução da função *updateAulasDadas()*, que é responsável por verificar se a data da aula é posterior à data atual, e assim sendo, essa aula é removida do vetor aulas e é adicionada ao vetor de aulas_dadas.

Esta classe é responsável por ler a informação dos ficheiros no ínicio da aplicação, convertendo-a nos respetivos objetos e guardá-la nos respetivos vetores. Por outro lado, no término da execução do programa guarda a informação contida nos vetores nos respetivos ficheiros. É também aqui que se encontram funções para adicionar e remover objetos dos vetores (acima mencionados), bem como funções que tratam dos vários tipos de *sorting* nos vetores que contêm a informação.

Para implementar a funcionalidade da loja, os artigos existentes são guardados numa árvore binária de pesquisa, ordenados pela sua designação e para a mesma designação por tamanho. Um artigo é definido pela sua designação, pelo seu tamanho, pelo seu preço e pela quantidade disponível (stock). Para isso, foi criada a classe Item correspondente a um artigo, cujos dados membro são a designação, o tamanho, o preço e o stock. Quando um funcionário compra ao fornecedor ou vende um artigo, este artigo é procurado na árvore binária e o seu stock é incrementado ou decrementado, respetivamente. Caso um artigo ainda não exista na árvore binária, é possível adicionar um novo nó para esse artigo, utilizando a opção “Adicionar um novo item.”.

Foi adicionado um novo dado membro à piscina do tipo pair <int, int> que contém as coordenadas relativas à localização da piscina. Consoante o tipo de aula pesquisada, caso esta não seja leccionada na piscina corrente, é consultada a fila de prioridade que contém as restantes piscinas ordenadas pela distância dessas piscinas à atual e é retornada a piscina que se encontrar mais próxima.

No ínicio da execução do programa, a priority_queue<Piscina> recebe todas as piscinas, organizando-se de modo a que a que esteja no topo é a mais perto da piscina atual. Para isto foram utilizadas duas variaveis de coordenadas: umas coordenadas reais e outras fictícias. As coordenadas reais são as coordenadas x e y da piscina. As coordenadas fictícias são calculadas sempre que o programa é iniciado, em funcao das coordenadas reais da piscina na qual se está a trabalhar.

Os utentes que não frequentem a piscina há mais de um determinado período de tempo (trinta dias) são colocados numa tabela de dispersão como inativos no dado membro tabHutente utentes_Idle. Foram adicionados à piscina os dados membro Data dia_atual e Data lastDayUsed responsáveis por guardar a data atual e a data da última utilização da piscina.

Na classe Utente foram adicionados os seguintes dados membro: telemóvel, mail e morada e o IdleDays que contém o número total de dias em que o utente se encontra ausente da piscina. Sempre que a aplicação é ligada, a variável IdleDays é atualizada conforme a diferença entre a data atual e a data da última utilização da piscina. Quando o valor de IdleDays é superior ao limite (trinta dias), o utente é colocado na tabela de dispersão e marcado como inativo. Sempre que uma aula passa para uma aula dada, todos os utentes que a frequentaram têm a sua variável IdleDays de volta a 0.

Para complementar o trabalho e seguindo o esquema da parte 1, agora cada piscina para além de coordenadas tem uma variavel que é atulizada sempre que se faz login na piscina. Esta variável regista a última data de acesso à piscina e serve para atualizar os IdleDays de cada utente, podendo assim fazer as devidas mudanças a nivel de utentes sempre que o programa é ligado.

Por fim também é feita uma campanha que entra em vigor todos os dias 1 de cada mês (após saldar as dívidas de cada utente), a todos os utentes inativos.

Diagramas de classes

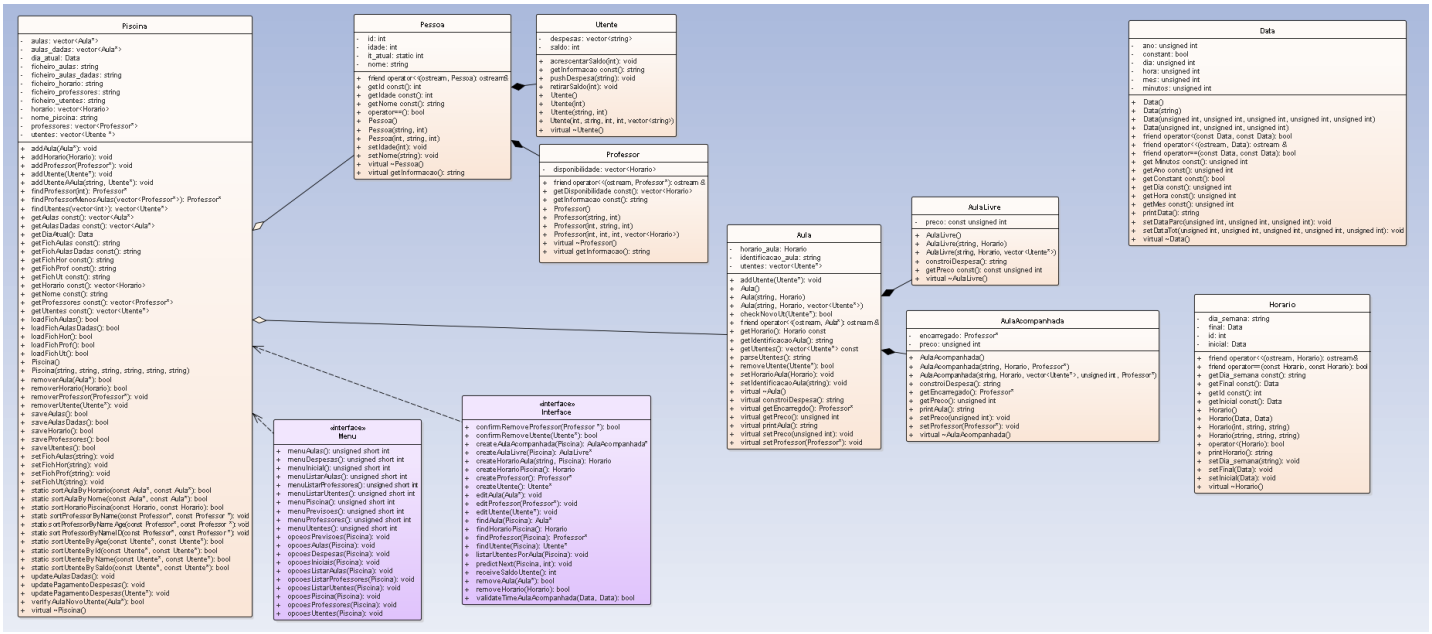

Diagrama com melhor resolução enviado no ficheiro .zip com nome DiagramaDeClasses.png

Lista de casos de utilização

Após o *login* ter sido efetuado, a consola mostra o Menu Inicial em que o utilizador pode selecionar uma das seguintes opções:

Menu Inicial

1. Menu Utentes
2. Menu Professores
3. Menu Aulas
4. Menu Piscina
5. Guardar e sair do programa

Como pretendemos dar a conhecer de forma detalhada os casos de utilização da aplicação, vamos seguir ordenadamente cada uma das opções existentes.

- **Menu Utentes**

Neste menu, estão presentes as operações que se podem realizar ao longo da utilização do programa, no que diz respeito aos utentes:

1. Criar Utente
2. Editar Utente
3. Remover Utente
4. Listar Utentes
5. Fazer um depósito
6. Remover saldo
7. Voltar ao menu inicial

1. Criar utente – é pedido ao utilizador que introduza nome e idade válidos e é guardado o utente criado no vector do tipo Utente.

2. Editar Utente – o utilizador tem a possibilidade de alterar os dados referidos no ponto anterior, depois de escolher o id do utente a editar.

3. Remover Utente – após selecionado o id do utente a eliminar, este é removido do vetor do tipo Utente.

4. Listar Utentes – como estão implementadas diversas formas de fazer a listagem, criamos um novo menu que apresenta as diferentes opções:

- **Menu Listar Utentes**
  1. Listar utentes por id
  2. Listar utentes por nome
  3. Listar utentes por idade
  4. Listar Utentes por saldo
  5. Voltar ao menu de utentes

Para fazer as diferentes ordenações no vetor de utentes, tornou-se indispensável o uso da função genérica *sort* disponível na *C++ Standard Library.* A utilização desta função exigiu que se implementassem vários operadores de comparação nas respetivas classes.

5. Fazer um depósito – escolhendo o id do utente, pode fazer-se um depósito na sua conta, incrementando o saldo disponível.

6. Remover saldo – é possível fazer o contrário do que foi especificado no ponto anterior, basta decrementar o saldo do utente selecionado.

7. Voltar ao menu inicial – o utilizador deixa o menu específico em que se encontra e volta para o anterior, mais geral.

- **Menu Professores**
  1. Criar Professor
  2. Editar Professor
  3. Remover Professor
  4. Listar Professores
  5. Voltar ao menu inicial

Neste menu, estão presentes as operações que se podem realizar ao longo da utilização do programa, no que diz respeito aos professores e que são muito semelhantes às que foram descritas para os utentes.

De referir que em 2. Editar Professor - também se mostra o respetivo horário do professor, para que o utilizador tenha a certeza que é esse professor que pretende editar.

Em 4. Listar Professores, surge outro menu para se proceder à listagem selecionada:

- **Menu Listar Professores**
  1. Listar Professores por id
  2. Listar Professores por nome
  3. Listar professores por idade
  4. Voltar ao menu de professores
  5. Voltar ao menu inicial – o utilizador deixa o menu específico em que se encontra e volta para o anterior, mais geral.

Terminadas as operações relativas aos professores, analisemos agora as hipóteses de seleção presentes no terceiro menu:

- **Menu Aulas**
  1. Adicionar Aula Livre
  2. Adicionar Aula Acompanhada
  3. Editar Aula
  4. Remover Aula

5. Listar Aulas
6. Adicionar utente a uma Aula
7. Remover utente de uma aula
8. Voltar ao menu inicial

1. Adicionar Aula Livre – pede-se ao utilizador que introduza o nome da nova aula livre, que terá uma duração máxima de duas horas (considerando períodos múltiplos de trinta minutos), bem como a sua data de ínicio no formato: ano-mês-dia-hora-minutos. O utilizador escolhe uma das seguintes quatro opções:

1. 30 minutos
2. 1 hora
3. 1 hora e trinta minutos
4. 2 horas

e se não existirem incompatibilidades com as restantes aulas, a aula livre é adicionada com sucesso.

2. Aula Acompanhada – tal como no ponto anterior, o utilizador digita o nome da nova aula livre, a data de ínicio e o preço desta aula (que tem sempre a duração de uma hora, e portanto, o seu preço é fixo).

3. Editar Aula – o utilizador seleciona o tipo de aula que pretende editar (aula livre ou aula acompanhada), digitando 1 ou 2, respetivamente. Posteriormente introduz o nome da aula que deseja alterar e de seguida altera-lhe o nome.

4. Remover Aula – o utilizador seleciona o tipo de aula que pretende eliminar (aula livre ou aula acompanhada), e introduz o nome da aula. No caso dessa aula existir, é-lhe mostrada uma mensagem de confirmação e basta digitar “y” para prosseguir com a operação.

5. Listar Aulas – o utilizador pode aceder à informação disponível acerca das aulas e obtê-la sob a forma de listagens:

- **Menu Listar Aulas**
  1. Listar Aulas por nome
  2. Listar Aulas por Horário
  3. Listar utentes de uma Aula
  4. Voltar ao menu de Aulas

6. Adicionar utente a uma Aula – após selecionar o tipo da aula e indicar o seu nome, pede-se ao utilizador que insira o id do utente a acrescentar nessa aula. O utente é adicionado ao vetor de utentes que frequentam a respetiva aula.

7. Remover utente de uma aula – o mesmo foi descrito anteriormente, desta vez com o prepósito de eliminar o utente selecionado do vetor de utentes da aula escolhida.

8. Voltar ao menu inicial – o utilizador deixa o menu específico em que se encontra e volta para o anterior, mais geral.

- **Menu Piscina**
  1. Adicionar Horário
  2. Remover Horários
  3. Listar Horários
  4. Menu Despesas
  5. Menu Previsões
  6. Voltar ao menu inicial

1. Adicionar Horário – o utilizador introduz o dia da semana em pretende definir o horário de funcionamento da piscina no formato hora-minutos.

2. Remover Horários – o utilizador pode remover o horário relativo a um dia da semana.

3. Listar Horários – podem ser vizualidados os horários existentes para cada dia da semana.

4. Menu Despesas – neste menu, o utilizador pode escolher entre pagar as despesas de um utente ou pagar as despesas totais:

- **Menu Despesas**
  1. Pagar despesas totais
  2. Pagar despesas de um utente
  3. Voltar ao menu Inicial

5. Menu Previsões – aqui podem ser consultadas as estatísticas da piscina.Nos diferentes períodos de tempo escolhidos, o utilizador recebe uma mensagem:

```
Neste periodo do tempo irao decorrer X aulas, que irao gerar Y euros e vao estar presentes W utentes.
```

Em que X é número de aulas previsto para esse período, Y o montante que essas aulas vão gerar e W é o número de utentes que vão frequentar essas aulas.

- **Menu Previsões**

1. Fazer previsão para amanhã
2. Fazer previsão para a semana
3. Fazer previsão para o próximo mês
4. Fazer previsão para o próximo ano

Com a atualização da aplicação, surgiram os seguintes Menus:

- **Menu Loja**
  1. Vender item
  2. Adicionar um novo item
  3. Abastecer um item
  4. Editar preco de um item
  5. Remover um item
  6. Fazer o inventario
  7. Voltar ao Menu Inicial

1. Vender Item – o funcionário digita o id do cliente que procederá à compra e são perguntadas ao utilizador, as características do produto a ser transaccionado; Se todas as informações estiverem corretas, sendo que o item terá de existir e ter um stock maior que 0, esse mesmo stock é decrementado por uma unidade e o saldo do utente que fez a compra e decrementado em funcao do preço do artigo.

2. Adicionar um novo item – O funcionario digita as informacoes relativas a um novo artigo (nome, tamanho e preco) e é criado um novo nó na BST com esse artigo.

3. Abastecer um item – o funcionário digita a descrição do produto e a quantidade que deseja abastecer e, se o artigo estiver na sua loja, é abastecido.

4. Editar preco de um item – o funcionário pode editar o preço de um determinado produto, após especificar o seu nome e tamanho.

5. Remover um item – Esta opção serve para remover completamente um item da BST, ou seja, o seu nó; é de notar que, se um funcionario quiser eliminar um determinado item completamente, tera de eliminar tantas vezes quanto o artigo aparece na BST, isto é, se existirem boias de tamanho 1 e boias de tamanho 2, o funcionario terá de eliminar ambas as boias.

4. Fazer o inventario – é possível fazer a listagem do inventário existente na loja.

- **Menu Encontrar**
    1. Encontrar piscina mais perto com uma aula
    2. Relatório de piscinas em função da aula
    3. Listar todas as piscinas por distância
    4. Voltar ao Menu Inicial

1. Encontrar piscina mais perto com uma aula – Nesta opção, o utilizador tem a oportunidade de escolher entre uma aula livre ou acompanhada, e a modalidade praticada nessa aula, sendo que a aplicação lhe retorna a piscina mais próxima em que determinada aula com as características especificadas pelo utilizador irá acontecer.

2. Relatório de piscinas em função da aula – Consoante as características da aula especificada, se esta existir, a aplicação informa o utilizador de todos os casos em todas as piscinas em que a aula irá decorrer.

3. Listar todas as piscinas por distância – É feita a listagem de todas as picinas, conforme a distância a que se encontrem da piscina atual.

- **Menu Idle**
    1. Campanha para todos os utentes inativos
    2. Remover todos os estados inativos
    3. Listar tabela de utentes inativos

1. Campanha para todos os utentes inativos – Aqui, os utentes identificados como inativos, têm a possibilidade de editarem a sua informação, de modo a que esta se encontre sempre atualizada na aplicação.

2. Remover todos os estados inativos – Nesta opcção, os utentes em estado inativo passam a ficar ativos novamente.

3. Listar tabela de utentes inativos – Podemos obter a listagem de todos os utentes inativos.

Dificuldades

Dado que o enunciado nos foi dado de forma muito abstrata, a primeira dificuldade surgiu aquando da decisão de quais seriam as estruturas que melhor se adaptariam à aplicação pretendida.

A utilização do novo IDE *Eclipse* causou alguns problemas ao nível de *debugging* da solução.

Sendo que a criação de diagramas *UML* e a documentação utilizando o *Doxygen* foram uma novidade, também nos causaram algumas dificuldades na sua utilização.

Ao nível do desenvolvimento do código, a parte mais complicada de fazer foram as funções que tratam da leitura e da escrita dos ficheiros.

Esforço dedicado

Com o esforço demonstrado por todos os elementos do grupo conseguimos superar as dificuldades e concluir assim o projeto com sucesso. O próprio tema foi bastante interessante, o que ajudou a não perder o ânimo. Seguidamente apresentamos as percentagens que correspondem ao empenho e dedicação mostrado na realização do trabalho:

Cristiana Ribeiro – 33%

Mário Fernandes – 33%

Sérgio Salgado – 33%