

Relatório Técnico do projecto ARIADNE

Praxis XXI

Guia Móvel do Lazer: Planeamento, Análise e Desenho

Carlos Correia
Norman Noronha
Daniel Gomes

Junho de 2000

INDICE

1	INTRODUÇÃO.....	3
1.1	MOTIVAÇÃO.....	3
1.2	PROPOSTO.....	4
2	TRABALHO DESENVOLVIDO.....	5
2.1	CONCRETIZADO E PROPOSTO.....	5
2.2	DESCRIÇÃO FUNCIONAL DO GUIA DO LAZER PERSONALIZADO.....	5
2.3	DESCRIÇÃO FUNCIONAL DO GUIA MÓVEL DO LAZER.....	5
2.3.1	<i>Aplicação Lazer para o Palm Pilot</i>	5
2.3.2	<i>Conduta de Sincronização</i>	6
3	CENÁRIOS E CASOS DE USO.....	7
3.1	GUIA DO LAZER PERSONALIZADO.....	7
3.2	GUIA DO LAZER MÓVEL.....	8
4	ESPECIFICAÇÃO DE REQUISITOS DE CONTEXTO.....	10
4.1	GUIA DO LAZER PERSONALIZADO.....	10
4.2	GUIA MÓVEL DO LAZER.....	10
4.2.1	<i>Aplicação Lazer para o Palm Pilot</i>	10
4.2.2	<i>Conduta/Sincronização</i>	11
5	AMBIENTE DE DESENVOLVIMENTO.....	12
5.1	CDKJ.....	12
5.2	METROWORKS CODE WARRIOR.....	12
5.3	BD ORACLE 8.....	12
6	ANÁLISE DE SISTEMA.....	13
6.1	MODELOS DE OBJECTOS.....	13
6.1.1	<i>BD Oracle</i>	13
6.1.2	<i>Palm Pilot</i>	15
7	DESENHO DE SISTEMA.....	16
7.1	ARQUITECTURA.....	16
7.2	MODELO DINÂMICO.....	17
8	PLANO DE TESTES.....	18
8.1	GUIA DO LAZER PERSONALIZADO.....	18
8.1.1	<i>Teste Funcional GUI</i>	18
8.2	APLICAÇÃO LAZER (PALMPILOT).....	19
8.2.1	<i>Testes Estruturais</i>	19
8.2.2	<i>Testes Funcionais</i>	24
8.3	CONDUTA.....	27
8.3.1	<i>Testes estruturais</i>	27
8.3.2	<i>Testes funcionais</i>	29
9	BUGS CONHECIDOS.....	30
10	CRITICA ÀS ESTIMATIVAS.....	31
11	CRITICA AO PLANEAMENTO E ANÁLISE DE RISCO.....	34
12	CONCLUSÕES E ANÁLISE CRITICA DO PROJECTO.....	36
13	TRABALHO FUTURO.....	37
14	BIBLIOGRAFIA.....	38

1 Introdução

1.1 Motivação

A realidade actual da massificação e banalização de utilização dos *palmtops* (dispositivos portáteis de pequena dimensão) deixa antever uma nova perspectiva de utilização destes dispositivos como meios preferenciais para o acesso à informação já disponível na Web.

O objectivo do presente projecto é o desenvolvimento de um protótipo de uma *information appliance* para o acesso ao Guia do Lazer, uma publicação electrónica do Público, com informação sobre eventos de lazer nas cidades de Lisboa e Porto (Inicialmente). Esta informação será acedida através de dispositivos de computação portáteis de pequena dimensão (dispositivos de bolso) nomeadamente, o *PalmPilot*.

A informação a divulgar já é actualmente disponibilizada na página Web do Público *on-line*, Guia do Lazer, nomeadamente, os eventos e locais onde ocorrem (por exemplo: concertos, exposições, peças de teatro, dança, ópera e festas na cidade de Lisboa e Porto).

Pretende-se construir um "Guia do Lazer Personalizado", a partir do Guia do Lazer já existente. Este permitirá aos seus utilizadores definirem as consultas realizadas regularmente quando acedem ao Guia e obterem essa informação automaticamente a partir da base de dados do Guia do Lazer. Esta informação pode ser anotada e pontuada (por exemplo: pontuação de um dado restaurante de comida chinesa).

A construção deste Guia do Lazer Personalizado para computadores de bolso obedece a dois objectivos fundamentais:

- acesso rápido às actualizações da informação consultada regularmente pelos utilizadores do Guia do Lazer, através da edição personalizada;
- alargamento do âmbito de utilização da informação do Guia do Lazer, de forma a que os utilizadores móveis possam aceder à informação pretendida durante períodos de desconexão.

1.2 Proposto

No final de 1999 decorreu um projecto com o jornal o Público no âmbito do projecto ARIADNE, com o objectivo de desenvolver um protótipo do Guia do Lazer personalizado para dispositivos de bolso.

Pretende-se agora desenvolver um projecto académico para a cadeira de PSI dentro do prazo dado (5 de Junho) com características de expansibilidade, de modo a permitir a sua viabilidade a nível comercial à posteriori.

Neste projecto propomos o desenvolvimento de:

- Guia do Lazer Personalizado a partir da publicação electrónica do Guia do Lazer do Público on-line já existente, com informação sobre eventos de lazer nas cidades de Lisboa e Porto.

Esta tarefa foi parcialmente desenvolvida no projecto anterior. Neste projecto pretende-se dar continuidade a este trabalho. De modo a tornar a aplicação robusta e com a qualidade de um produto final, é necessário efectuar alguns melhoramentos nos módulos que apresentam níveis deficientes de desempenho e usabilidade.

- Guia Móvel do Lazer - sistema de acesso ao Guia do Lazer no PalmPilot, contendo os seguintes módulos:
 - Visualização e edição de informação relativa a restaurantes;
 - Visualização e edição de informação relativa a cinemas;

Os módulos devem permitir que a informação sobre cinemas e restaurantes seja pontuada.

- Condução para sincronização dos dados existentes na base de dados Oracle com o sistema de visualização no Palm.

Esta separação do projecto em duas partes não é casual, mas imposta pela diferença de características entre elas, pelo que será seguida ao longo deste relatório de modo a proporcionar ao leitor uma explicação o mais explícita possível.

2 Trabalho Desenvolvido

2.1 Concretizado e proposto

Como se verificou uma sobrecarga de trabalho imposta não só pelo desenvolvimento deste projecto mas também por projectos de outras cadeiras, não foi possível a concretização de todas as funcionalidades inicialmente propostas.

Assim sendo optámos por desenvolver um sistema focado apenas na informação relativa aos Restaurantes, uma vez que esta restrição não causa uma grande simplificação do sistema do ponto de vista arquitectural ou algorítmico, mas apenas redução da quantidade de código a ser gerado, que toma dimensões consideráveis para funções aparentemente simples, na codificação da aplicação desenvolvida para o Palm, uma vez que é desenvolvida numa linguagem de baixo nível(C) e numa plataforma com características peculiares .

2.2 Descrição funcional do Guia do Lazer Personalizado

O Guia do Lazer Personalizado permite ao utilizador aceder a dados sobre lazer de uma forma personalizada na Internet. Para tal este Guia fornece ao utilizador um formulário de registo e um de personalização de perfil. Sendo que, o formulário de registo é preenchido apenas uma vez, para que o utilizador fique com uma ficha no servidor.

O formulário do perfil pode ser alterado quando o utilizador desejar, este formulário serve para o utilizador escolher que tipo de informação pretende visualizar quando aceder a esta pagina.

Tendo os dois formulários preenchidos, sempre que o utilizador aceder ao URL da página terá ao seu dispor a informação que escolheu sem ter que a procurar.

2.3 Descrição funcional do Guia Móvel do Lazer

2.3.1 Aplicação Lazer para o Palm Pilot

A aplicação interage com o utilizador através de um sistema de janelas e menus, que obedecem às regras de interface do Palm Pilot. devidas às características particulares do dispositivo das quais se destacam as dimensões reduzidas do ecran, reduzida capacidade de processamento e armazenamento.

Os registos dos restaurantes podem classificar-se em “privados” e “publicos” que passamos a explicar:

Publicos: são os registos recebidos da Base de Dados Central quando um utilizador efectua uma sincronização.

Privados: são os registos criados ou alterados pelo utilizador.

A aplicação disponibiliza as seguintes funcionalidades:

- escolha da cidade da qual pretende visualizar a informação acerca dos restaurantes(de momento só se encontram disponíveis as cidades de Lisboa e Porto).
- visualização de restaurantes públicos, privados ou ambos
- edição fácil dos dados.
- configuração da visualização de forma a facilitar a pesquisa de dados.
- pesquisa rápida de restaurante a partir do nome
- notificação da Base de Dados Central de incorrecções nos dados dos restaurantes “públicos”.
- votação sobre a classificação geral de um restaurante

2.3.2 Conduta de Sincronização

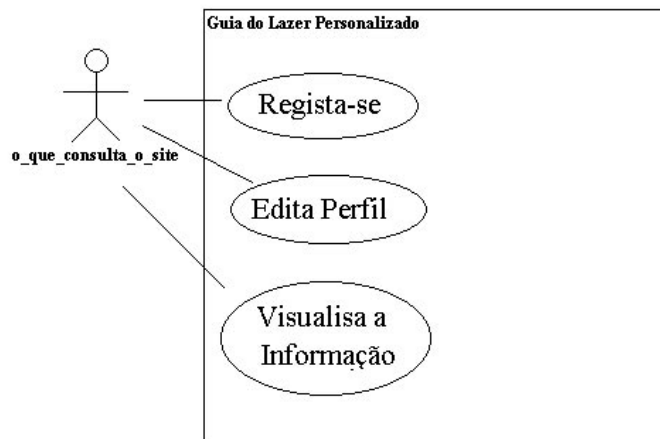
A Conduta é um programa cuja função é estabelecer a comunicação entre a Base de Dados e o Palm Pilot efectuando transferência e manipulação de dados entre os intervenientes quando se dá o processo de sincronização (Hot Sync).

A Conduta recebe da Base de Dados Central um conjunto de registos, definido pelo seu administrador, com informação acerca dos restaurantes, encarregando-se de processar a informação de modo a enviá-la num formato legível pela aplicação Lazer do Palm Pilot.

No sentido inverso da sincronização, recolhe as notificações das alterações efectuadas pelos utilizadores aos restaurantes Publicos e regista-as num Log da Base de Dados Central, sendo esta informação utilizada pelo administrador para fins de manutenção.

3 Cenários e casos de uso

3.1 Guia do Lazer Personalizado



Caso de Uso : Regista-se

Cenário A : Tem início quando o Actor acede a página Lazer Personalizado, o Actor activa o link registe-se aqui, o sistema por sua vez fornece ao Actor um formulário de registo. O Actor preenche todos os campos correctamente e pressiona o botão de enviar. O sistema verifica todas as entradas. O utilizador está registado.

Cenário B : (Alternativa ao Cenário A após o sistema fornecer o formulário de registo).

O Actor não preenche correctamente os campos do formulário e pressiona o botão de envio . O sistema informa o Actor de quais os campos que não estão correctos e devolve-lhe o formulário de registo.

Caso de Uso : Edita Perfil

Cenário A : Este cenário tem início quando o Actor “registado” activa o link editar perfil.

O sistema fornece-lhe um formulário de edição de perfil . O Actor preenche todos os campos os campos correctamente e pressiona o botão de enviar . O sistema verifica todas as entradas. O perfil está editado.

Cenário B : (Alternativa ao Cenário A após o sistema fornecer o formulário de edição de perfil).

O Actor não preenche correctamente os campos do formulário e pressiona o botão de envio . O sistema informa o Actor de quais os campos que não estão correctos e devolve-lhe o formulário de edição de perfil.

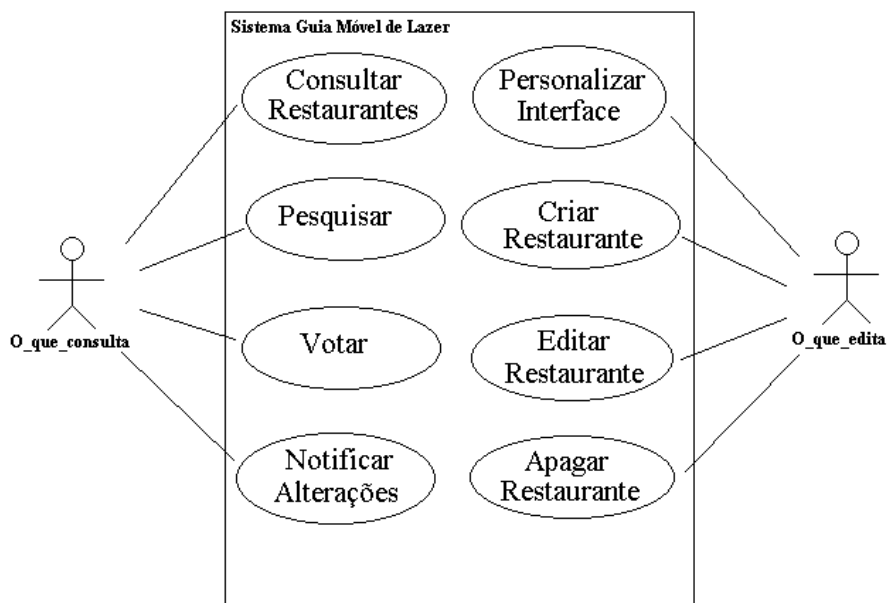
Caso de Uso : Visualização de Informação

Cenário A : O Actor acede à pagina do Lazer Personalizado pela primeira vez , regista-se no formulário de registo, edita o seu perfil no formulário correspondente e no fim destas duas operações o sistema fornece ao Actor a informação segundo os parâmetros que ele estabeleceu na personalização de perfil.

Cenário B : O Actor acede à pagina do Lazer Personalizado pela segunda vez , utilizando o mesmo browser que utilizou da primeira. O sistema pede informações ao browser e envia para o utilizador a página com as características que o utilizador estabeleceu no seu perfil.

Cenário C : O Actor acede à pagina do Lazer Personalizado pela segunda vez, utilizando outro browser. O Actor preenche os campos de Identificador e Password em seguida pressiona o botão de enviar. O sistema verifica os dados e envia para o utilizador a pagina com as características que o utilizador estabeleceu no seu perfil.

3.2 Guia do Lazer Móvel



Caso de Uso : **Consultar Restaurantes**

Cenário A : O Actor começa por seleccionar a palavra Restaurantes, o sistema fornece novo menu, onde o actor irá escolher de que cidade pretende visualizar os restaurantes, bastando para tal pressionar a área do ecrã onde se encontra o nome da cidade que deseja. Em seguida será apresentada no ecrã uma serie de restaurantes, de novo o Actor pressiona o nome do restaurante que deseja. Serão então mostradas todas as informações disponíveis sobre o restaurante seleccionado.

Caso de Uso : **Pesquisar**

Cenário A : Para fazer uma pesquisa por nome basta que quando estiver no menu restaurantes, o Actor coloque o cursor no tracejado em frente à palavra PESQUISAR e comece a escrever o nome do restaurante que deseja visualizar.

Caso de Uso : **Votar**

Cenário A : O actor vota num restaurante. O actor vai até ao menu restaurantes, escolhe o restaurante que deseja votar, o sistema fornece ao actor o menu com detalhes sobre o restaurante nesse menu, o actor pressiona o botão com a palavra votar. Será em seguida dada ao utilizador uma escala e um botão de continuar e outro de anular, o actor pressiona primeiro o valor que deseja atribuir ao restaurante e depois o botão continuar.

Caso de Uso : **Personalizar Interface**

Cenário A : Para personalizar as colunas. O actor carrega no botão de menus do Palm e aparece no topo a palavra “colunas”, o utilizador carrega na palavra colunas. Em seguida o actor pode escolher quais os campos que quer ver nas duas colunas disponíveis e por que campo quer os resultados ordenados para isso basta pressionar os valores correctos nas drop boxes e depois pressionar continuar.

Caso de Uso : **Criar Restaurante**

Cenário A : O actor quer criar um novo restaurante. Vai até ao menu Restaurantes e pressiona o botão novo, preenche os campos que se encontram no ecrã e pressiona continuar.

4 Especificação de requisitos de contexto

4.1 Guia do Lazer Personalizado

- Na primeira visita ao site o utilizador têm de se registar e editar o perfil.
- Para editar o perfil o utilizador necessita já estar registado no site.
- Quando um utilizador não utiliza o seu browser habitual (através do qual se registou) necessita introduzir o seu id e password.

4.2 Guia Móvel do Lazer

4.2.1 *Aplicação Lazer para o Palm Pilot*

- Um registo de um restaurante contém informação relativa à sua morada, zona, telefone, horário, cozinha, preço médio, descanso, classificação dos utilizadores e se aceita cartão de crédito,.
-
- Os restaurantes “públicos” são os recebidos da Base de Dados Central após uma sincronização, estando disponíveis a todos os utilizadores do sistema.
- Os restaurantes privados são os que sofreram alguma alteração localmente por um utilizador. Um restaurante “público” que tenha sido alterado ou um novo registo de um restaurante que tenha sido criado pelo utilizador torna-se restaurante “privado”.
- Os registos da informação relativa aos restaurantes só podem ser públicos ou privados.

4.2.2 Conduta/Sincronização

- Apenas os utilizadores registados podem sincronizar o seu Palm Pilot com a Base de Dados Central.
- O utilizador deverá introduzir o login/password que lhe foi atribuído quando se registou no Palm Pilot. Esta informação ser-lhe-á pedida uma única vez quando lançar a aplicação Lazer pela primeira vez no seu Palm.
- Antes de um utilizador efectuar uma sincronização deverá indicar à Base de Dados Central que o pretende fazer, após esta indicação ficará autorizado a efectuá-la.
- Quando a sincronização for efectuada os registos guardados na base de dados "Publico" do Palm Pilot serão substituídos pelos recebidos da Base de Dados Central.
- Os registos enviados pela Base de Dados Central são definidos pelo seu administrador.
- Quando um utilizador altera um registo no seu Palm, este é passado para a base de dados local "Privada", de modo a que o utilizador tenha sempre acesso às suas alterações/votações.
- Quando se dá o processo de sincronização a Base de Dados não tem acesso aos dados pessoais dos utilizadores, apenas é notificada, caso o utilizador o deseje, de que houve uma alteração efectuada a um dos seus registos. De modo a que através da análise destas informações seja possível uma manutenção mais eficiente da BD.
- As alterações aos registos efectuadas pelos utilizadores deverão ser registadas numa tabela de Log acessível à administração do sistema.
- O utilizador só deverá notificar a Base de Dados Central se tiver encontrado alguma incoerência na informação recebida desta.
- Os registos criados pelo utilizador são sua propriedade privada, logo durante a sincronização estes são "invisíveis"

5 Ambiente de desenvolvimento

5.1 CDKJ

A sincronização entre os dados do PALM e uma Base de Dados são efectuados através de uma conduta. Para podermos compreender melhor esta sincronização introduzimos em seguida o conceito de conduta e o ambiente de desenvolvimento CDKJ (Conduit Development Kit for Java).

As condutas residem no PC e são executadas pelo HOTSYNC Manager Program quando o utilizador liga o seu PALM ao PC através do “cradle” e ordena que se execute a sincronização pressionando o botão de HOTSYNC. A função da conduta é controlar a transferência de dados que ocorre entre os intervenientes na sincronização, manipulando e alterando informação consoante os propósitos desejados.

O CDKJ permite o desenvolvimento de condutas recorrendo à linguagem JAVA acrescentando um conjunto de classes que permitem a comunicação com o Palm Pilot, estendendo a norma que recorre ao C/C++.

5.2 Metroworks Code Warrior

Uma aplicação para o PalmPilot deve ser escrita na linguagem C com chamadas de funções ao sistema operativo do Palm(Palm Operativo System) e baseada em programação orientada por eventos.

O Metroworks CodeWarrior é uma aplicação que oferece suporte para a criação, compilação e “debugging” de aplicações para o PalmPilot, incluindo o Metroworks Constructor que consiste numa aplicação que ajuda a criação dos objectos da interface e a alteração aos seus atributos.

A documentação relevante às chamadas de funções do PalmOS, documentação de ajuda, um tutorial , exemplos e o ultimo emulador POSE para testar as aplicações encontram-se incluídos nesta ferramenta.

5.3 BD Oracle 8

Oracle 8.0.4. é um Sistema de Gestão de Bases de Dados que nos permite a criação e manutenção de bases de dados relacionais e interacção com aplicações escritas noutras linguagens através de bibliotecas.

6 Análise de sistema

6.1 Modelos de objectos

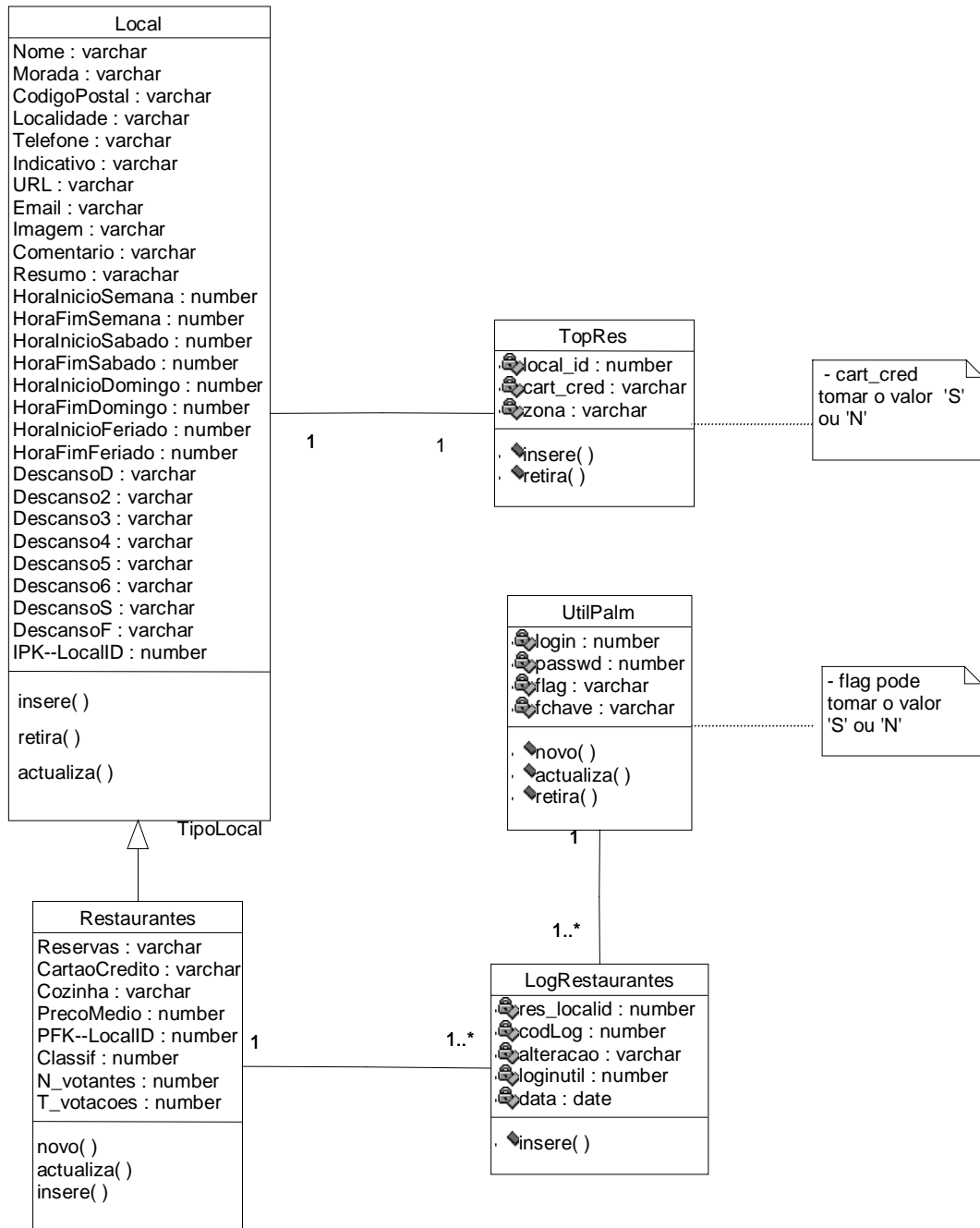
6.1.1 BD Oracle

6.1.1.1.1 Modelo de dados integrado no Guia do Lazer Personalizado



A tabela Utilizador guarda as informações pessoais do utilizador e a tabela perfil as suas preferências de modo a permitir o acesso rápido do utilizador à informação desejada.

6.1.1.1.2 Modelo integrado no Sistema Guia Móvel do Lazer



Este modelo de dados pertence a um esquema mais vasto que é partilhado por outros sistemas, como tal fomos instruídos para não alterar as tabelas existentes, antevendo-se uma reestruturação à posteriori que teria em consideração todas as alterações implicadas.

No entanto, apresentamos uma solução para a reestruturação do modelo, considerando que este seria apenas usado pelo nosso sistema:

- A tabela TopRes serve exclusivamente para determinar os tuplos a enviar para o Palm quando se dá a sincronização, assim sendo esta seria eliminada sendo substituída por uma View para cada cidade, sendo os seus atributos (cart_cred e zona) acrescentados à tabela Local, uma vez que se tratam de características. de qualquer local

6.1.2 Palm Pilot

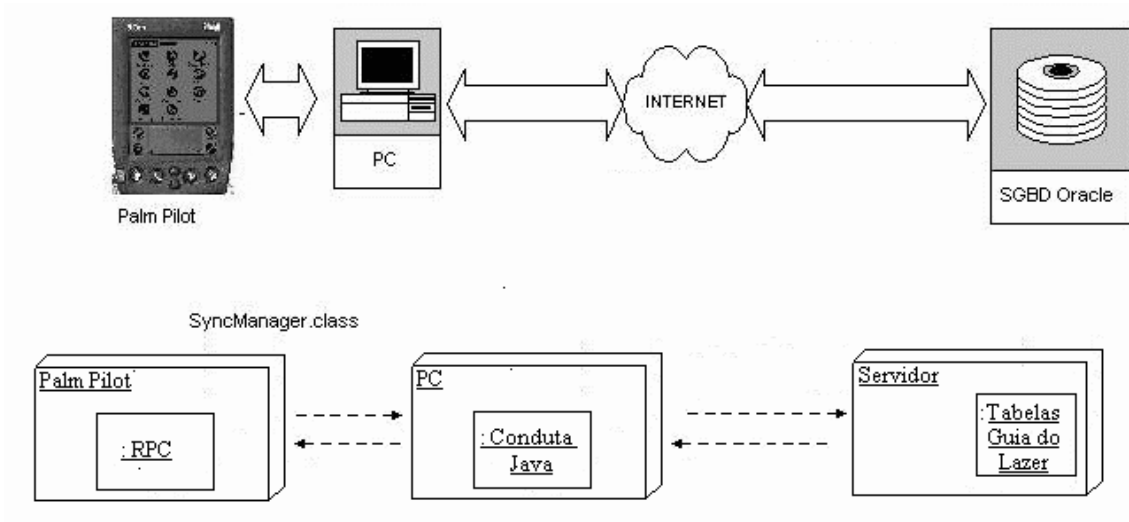
Restaurante
Nome:String Morada:String Zona:String Telefone:String Horario:String Cartão:String Cozinha:String Preço:String Descanço:String Classificação:String
Novo Alterar Consultar Eliminar

A informação relativa aos restaurantes de cada cidade é guardada numa tabela com o modelo de dados aqui apresentado.

Existindo uma tabela Publico e uma Privado para cada uma das cidades, ambas têm a mesma estrutura mas são manipuladas pela aplicação Lazer de maneiras diferentes

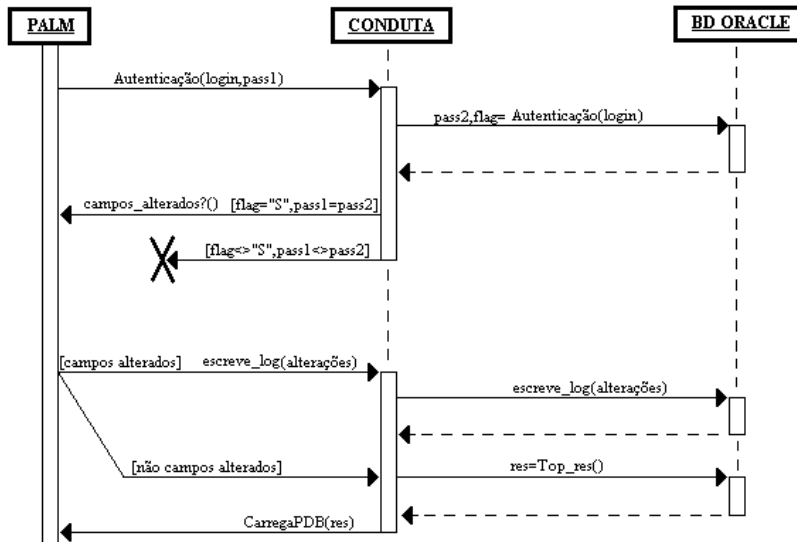
7 Desenho de sistema

7.1 Arquitectura



O sistema compreende pelo menos 3 componentes de Hardware e 3 componentes de Software que interagem entre si para criar o Guia do Lazer Móvel.

7.2 Modelo Dinâmico



O Palm dá início ao processo de sincronização enviando a sua informação de autenticação (login, password) à conduta, esta por sua vez através do login do utilizador recolhe a sua password e a flag de autorização de sincronização da BD Central (Oracle).

Se as duas passwords forem iguais e a flag for igual a 's', a conduta recolhe os registos alterados do Palm, analisando os campos que foram alterados para em seguida actualizar o Log da Base de Dados Central.

Quando não existirem mais registos com campos alterados a conduta pede à BD que lhe envie os registos dos restaurantes pré definidos, sendo esta informação transferida para a base de dados local do Palm (pdb).

8 Plano de Testes

8.1 Guia do Lazer Personalizado

8.1.1 Teste Funcional GUI

Teste das janelas:

- A janela abre correctamente depois de receber input do teclado ou qualquer menu? *SIM*.
- Tamanho da janela pode ser alterado? *SIM*.
- Será que todos os dados da janela possam ser acedidos pelo interface? *SIM*.
- A janela fecha bem e reabre conforme esperado? *SIM*
- Todas as funções relacionadas com a janela estão disponíveis? *SIM*
- Todas as funções relacionadas com a janela funcionam? *SIM*
- Todos os menus, botões encontram-se bem visíveis? *SIM*
- A janela fecha bem? *SIM*

Teste dos menus

- Os nomes do menu são auto explicativos? *SIM*
- Existe ajuda para cada menu? *NÃO* Os menus são extremamente simples, não necessitam qualquer ajuda.

Teste da entrada de dados

- A entrada de dados alfanuméricos são bem tratados e reaparecem no ecrã? *SIM*.
- Os modos gráficos de entradas de dados funciona correctamente? *SIM*.
- Dados inválidos são reconhecidos? *SIM*, existem vários testes que garantem uma coerência nos dados inseridos.

8.2 Aplicação Lazer (PalmPilot)

8.2.1 Testes Estruturais

1º Teste

```
static void MainFormInit(void) {  
    FormPtr frm;  
    TablePtr table;  
    UInt row;  
    CharPtr Cine = "Cinema";  
    CharPtr Rest = "Restaurante";  
    CharPtr space = " ";  
    FontID curFont;  
    RectangleType bounds;
```

```
    curFont = FntSetFont (largeBoldFont);  
    frm = FrmGetActiveForm();  
    table = FrmGetObjectPtr (frm, FrmGetObjectIndex (frm, LazerMainLazerMainTable));  
    TblSetColumnUsable (table, 0, true);
```

```
    for (row = 0; row < 2; row++) {
```

```
        TblSetRowUsable (table, row, true);  
        TblSetRowID (table, row, row);  
        TblMarkRowInvalid (table, row);  
        TblSetItemStyle (table, row, 0, labelTableItem);  
        TblSetItemPtr (table, row, 0, space);  
        TblSetItemInt (table, row, 0, row);  
        TblGetItemBounds (table, row, 0, &bounds);  
        TblSetRowHeight(table, row, 40);
```

```
        switch (row) {  
        case 0:
```

```
            WinDrawChars(Cine, StrLen(Cine), bounds.topLeft.x, bounds.topLeft.y);  
            break;
```

```
        case 1:
```

```
            WinDrawChars(Rest, StrLen(Rest), bounds.topLeft.x, bounds.topLeft.y);  
            break;
```

```
    }
```

```
    FntSetFont (curFont);  
}
```

1

2

3

4

5

6

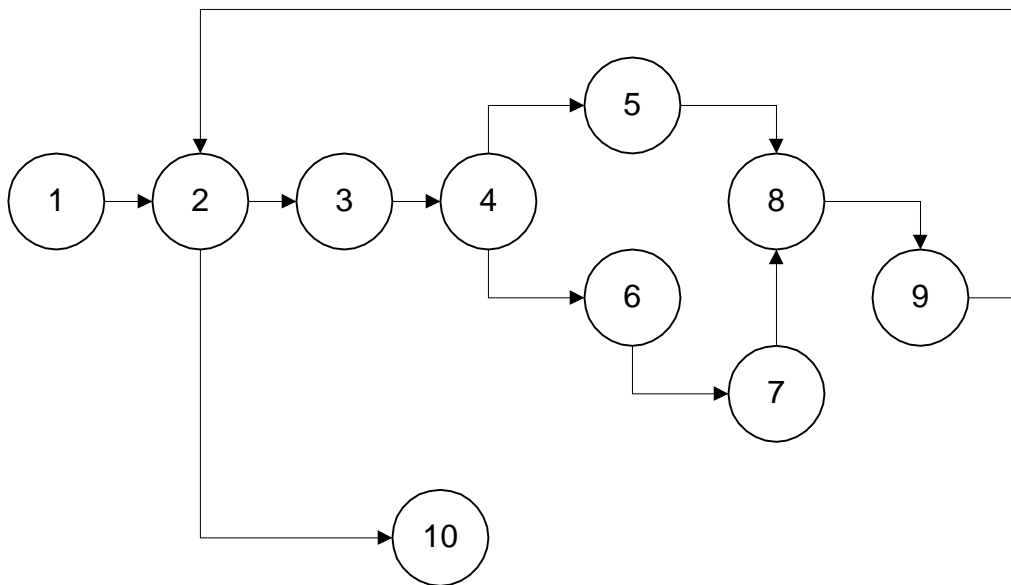
7

8

9

10

Grafo de fluxo:



Calculo de complexidade ciclomática:

A complexidade ciclomática é igual ao número de regiões independentes do grafo de fluxo, ou seja, 4.

Identificação de caminhos independentes:

- 1: 1-2-10
- 2: 1-2-3-4-5-8-9-2-10
- 3: 1-2-3-4-6-7-8-9-2-10
- 4: 1-2-3-4-6-8-9-2-10

Elaboração de casos de teste:

MainFormInit é uma função que desenha o objecto da tabela da página Main

Caso de teste 1: row: inválido

Resultado Esperado:

Tabela não é desenhada

Resultado Obtido: Confirmado

Caso de teste 2: row = 0

Resultado Esperado:

Primeira linha da tabela é desenhada

Resultado Obtido: Confirmado

Caso de teste 3: row = 1

Resultado Esperado:

Segunda linha da tabela é desenhada

Resultado Obtido: Confirmado

Caso de teste 4: row < 0

Resultado Esperado:

Tabela não é desenhada

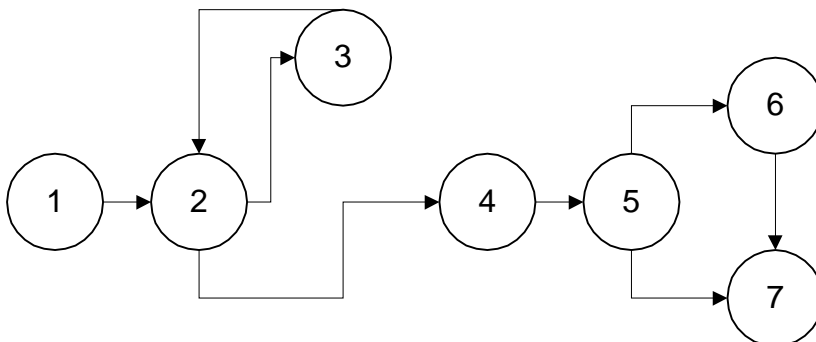
Resultado Obtido: Confirmado

2 teste

```
static void RestEditNewRecord (void) {  
    RestDBRecordType newRecord;  
    UInt i;  
    Err err;
```

```
    ① mode = pri;  
    ② for (i = 0; i < restNumFields; i++) { ③  
        newRecord.fields[i] = NULL;  
    ④ }  
    err = RestNewRecord(Lazer2DB, &newRecord, &CurrentRecord);  
    ⑤ if (err){  
    ⑥     ErrDisplay("RestEditNewRecord: Device Full");  
        return;  
    ⑦ }  
}
```

Grafo de fluxo :



Calculo de complexidade ciclomática:

A complexidade ciclomática é igual ao número de regiões independentes do grafo de fluxo, ou seja, 3

Identificação de caminhos independentes:

- 1: 1-2-3-4-5-7
- 2: 1-2-3-4-5-6-7
- 3: 1-2-4-5-7

Elaboração de casos de teste:

RestNewRecord é uma função que inicializa uma estrutura vazia para pedir um novo record.

Caso de teste 1: $I < \text{restNumFields}$ e $\text{err} == \text{NULL}$

Resultado Esperado:

Novo record criado

Resultado Obtido: novo record criado

Caso de teste 2: $I < \text{restNumFields}$ e $\text{err} \neq \text{NULL}$

Resultado Esperado:

Novo record não foi criado

Resultado Obtido: novo record não foi criado

Caso de teste 3: $I > \text{restNumFields}$ e err qualquer

Resultado Esperado:

Novo record não foi criado

Resultado Obtido: novo record não foi criado

8.2.2 Testes Funcionais

8.2.2.1 Teste da Interface gráfica do utilizador (GUI)

Teste das janelas:

- A janela abre correctamente depois de receber input do teclado ou qualquer menu? *SIM*.
- Tamanho da janela pode ser alterado? *NÃO* mas pode deslizar para baixo(scroll)
- Será que todos os dados da janela possam ser acedidos pelo interface? *SIM*.
- A janela fecha bem e reabre conforme esperado? *SIM*
- Todas as funções relacionadas com a janela estão disponíveis? *SIM*
- Todas as funções relacionadas com a janela funcionam? *SIM*
- Todos os menus, botões encontram-se bem visíveis? *NÃO* mas qualquer acesso aos menus no Palm é restrito, impondo-se a pressão da tecla específica do Menu para o seu aparecimento.
- A janela fecha bem? *SIM*

Teste dos menus

- Será que os menus “pull down” funcionam correctamente? *SIM*
- É possível invocar um comando do menu com o atalho de texto? *SIM*
- Os nomes do menu são auto explicativos? *SIM*
- Existe ajuda para cada menu? *NÃO*

Teste da entrada de dados

- A entrada de dados alfanuméricos são bem tratados e reaparecem no ecrã? *SIM*.
- Os modos gráficos de entradas de dados funciona correctamente? *SIM*.
- Dados inválidos são reconhecidos? *NÃO*, o utilizador poderá entrar quaisquer tipo de dados porque é do seu uso pessoal e nunca se realizará nenhuma computação directa sobre estes valores.

8.2.2.2 Classes de Equivalência

```
static Word RestNextRecord(UInt recpublic, UInt recprivate) {
    RestPackedDBRecord *src;
    UInt    newIndex;
    Handle  recordH;
    UInt    numRecordsPri, numRecordsPub;

    numRecordsPub = DmNumRecords(Lazer1DB);
    numRecordsPri = DmNumRecords(Lazer2DB);

    if (recpublic >= numRecordsPub && recprivate >= numRecordsPri) {
        return err;
    } else if (recpublic < numRecordsPub && recprivate < numRecordsPri) {
        recordH = DmQueryRecord(Lazer1DB, recpublic);
        src = (RestPackedDBRecord *) MemHandleLock(recordH);
        newIndex = RestFindSortPosition(Lazer2DB, src);
        MemPtrUnlock(src);
        if (newIndex <= recprivate) {
            return pub;
        } else {
            return pri;
        }
    } else {
        if (recpublic >= numRecordsPub)
            return pri;
        else
            return pub;
    }
}
```

8.2.2.2.1 Identificação de classes de equivalência

Identificação das classes de equivalência

- Parâmetro recpublic:

Classes de valores válidos:

P1 = {recpublic | restaurante com id existente na base de dados publica}

Classes de valores inválidos:

P2 = {recpublic | restaurante com id não existente na base de dados publica}

- Parâmetro recprivate:

Classes de valores válidos:

S1 = {recprivate | restaurante com id existente na base de dados privado}

Classes de valores inválidos:

S2 = {recprivate | restaurante com id não existente na base de dados private}

Total: 4 classes de equivalência

Testes de classes de equivalência

Teste Forte: $2^2 = 4$

Teste Fraco: 2

Testes de valores fronteira

Caso	Classe de Equivalência	Resultado Esperado	Resultado Obtido
1	pub = 0, pri E S1	o mais pequeno	valor confirmado
2	pub = 1, pri E S1	o mais pequeno	valor confirmado
3	pub = maxRecords - 1, pri E S1	o mais pequeno	valor confirmado
4	pub = maxRecords, pri E S1	o mais pequeno	valor confirmado
5	pub = maxRecords + 1, pri E S1	Err	valor confirmado
6	pri = 0, pub E P1	o mais pequeno	valor confirmado
7	pri = 1, pub E P1	o mais pequeno	valor confirmado
8	pri = maxRecords - 1, pub E P1	o mais pequeno	valor confirmado
9	pri = maxRecords, pub E P1	o mais pequeno	valor confirmado
10	pri = maxRecords + 1, pub E P1	Err	valor confirmado

Nota: maxRecords é o índice do ultimo record na base de dados

8.3 Conduta

8.3.1 Testes estruturais

```
/**  
Recebe a classificacao dada por um utilizador e actualiza  
a BD, alterando o t_votacoes, n_votantes e classif  
*/  
public void actualizaClassif( int classif, int localid)  
    throws SQLException , VotacaoImpossivelException{
```

1

```
int n_votantes = 0, t_votacoes = 0, novaclassif = 0;  
ResultSet val_act = null;  
Statement statement = conn.createStatement();  
Statement statement2 = conn.createStatement();  
  
//select for update  
val_act = statement.executeQuery("select n_votantes,  
                                t_votacoes from Restaurantes "+  
                                "WHERE localid = " + localid );
```

2

```
if(val_act.next()){
```

```
    n_votantes = val_act.getInt("n_votantes") + 1;  
    t_votacoes = val_act.getInt("t_votacoes") + classif;  
    novaclassif = t_votacoes / n_votantes;
```

3

4

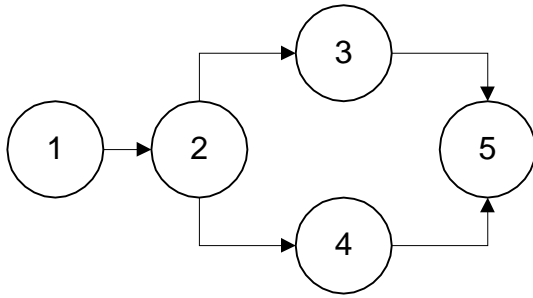
```
} else  
    throw new VotacaoImpossivelException("Votacao  
    impossivel: o restaurante n,,o existe na BD");
```

5

```
statement2.execute("UPDATE Restaurantes SET n_votantes= "  
                  + n_votantes + " , "+  
                  " t_votacoes = " + t_votacoes + " , "+  
                  " classif = "+ novaclassif +  
                  " WHERE localid = " + localid);  
  
statement.close();  
statement2.close();  
conn.commit();
```

```
}
```

Grafo de fluxo :



Calculo de complexidade ciclomática:

A complexidade ciclomática é igual ao número de regiões independentes do grafo de fluxo, ou seja, 2

Identificação de caminhos independentes:

1: 1-2-3-5

2: 1-2-4-5

Elaboração de casos de teste:

Actualizaclassif é uma função que calcula a nova classificação e altera o base de dados.

Caso de teste 1: localld: valido , classif qualquer

Resultado Esperado:

- 1) Val_act.next() != NULL
- 2) nvotantes = nvotantesOld + 1;
- 3) t_votações = tvotacoesOld + classif;
- 4) novaclassif = tvotacoes/novaclassif;

Resultado Obtido: Confirmado

Caso de teste 2: localld: invalido , classif qualquer

Resultado Esperado:

- 5) Val_act.next() == NULL
- 6) nvotantes = 0
- 7) t_votações = 0
- 8) novaclassif = 0

Resultado Obtido: Confirmado

8.3.2 Testes funcionais

8.3.2.1 Classes de Equivalência

Identificação das classes de equivalência

- Parâmetro localId:

Classes de valores válidos:

L1 = {localId | restaurante com localId existe na base de dados}

Classes de valores inválidos:

L2 = {localId | restaurante com localId não existe na base de dados}

Testes por classes de equivalência

Caso	Classe de Equivalência	Resultado Esperado	Resultado Obtido
1	I E L1	Nvotantes.tvotacoes e novaclassif são preenchidos.	valor confirmado
2	I E L2	Nvotantes, tvotacoes e novaclassif são todos 0	valor confirmado

Testes por valores fronteira

Caso	Classe de Equivalência	Resultado Esperado	Resultado obtido
1	l = -1	Nvotantes.tvotacoes e novaclassif são todos 0	valor confirmado
2	l = 0	Nvotantes+=1 tvotacoes+=1 novaclassif=tvotacoes/nvotantes	valor confirmado

9 BUGS conhecidos

Aplicação Lazer para o PalmPilot

Após uma votação, a janela de votação volta aparecer se a próxima operação for de mudar de categoria de restaurantes ou subir e descer na tabela dos restaurantes.

A pesquisa rápida para o caso de todos os restaurantes apresenta algumas limitações.

Sincronização

- Quando se efectua a sincronização as flags dirty do Palm deveriam ser limpas, no entanto quando a função de biblioteca da conduta que efectua esta operação é chamada, o Data Manager do Palm Pilot rebenta. Assim sendo, esta função foi desactivada causando a anomalia de cada vez de que um utilizador se sincroniza usando a conduta desenvolvida, as suas notificações são repetidas no Log da Base de Dados Central.

10 Crítica às Estimativas

Em primeiro lugar apresentamos um resumo das estimativas realizadas no 1º Relatório, nestas consideramos um valor de produtividade média igual a 1338LC/mês.

Estimação baseada em linhas de código

Componente	Estimativa Optimista (LC)	Estimativa Provável (LC)	Estimativa Pessimista (LC)	Valor Esperado (LC)
Aplicação para o Palm Pilot	2200	2800	3200	2766
BD Oracle	125	150	200	154
Interface Servidor (RMI)	300	400	600	416
Condução de Sincronização	600	700	900	716
Alterações ao Guia do Lazer Person.	325	400	500	404
			LC esperadas =	4456

Esforço = 3.3 pm

Duração = 1.11 meses

Estimação baseada em pontos de função

LC = 8667 linhas

Esforço = 6.4 pm

Duração = 2.1 meses

Modelo simples (COCOMO)

LC = 4456 linhas

Esforço = 11.52 pm

Duração = 6.32 meses

Modelo Intermédio

Esforço = 15,2 pm

Duração = 5 meses

Resultados obtidos

Após a finalização do projecto obtivemos os seguintes resultados:

Componente	Linhas de código
Aplicação para o Palm Pilot	5000
BD Oracle	130
Condução de Sincronização	1340
Alterações ao Guia do Lazer Person.	1300
Total	7770

Nota: Grande parte do código que faria parte da Interface Servidor foi incluído na Condução de sincronização

Esforço despendido = 7770/1338 = 5,8 pm

O elevado número de LC gerado deveu-se principalmente à codificação da aplicação para o Palm Pilot, (quase o dobro do valor esperado na estimativa por linhas de código), pois devido à falta de formação em relação à plataforma de desenvolvimento, subestimamos a sua complexidade e dimensão.

As linhas de código necessárias para a alteração do Guia do Lazer Personalizado também excederam largamente as expectativas, uma vez que grande parte do trabalho que era suposto alterado teve de ser refeito.

Análise de LC

Por comparação podemos concluir que a estimativa que se aproximou mais dos resultados obtidos foi a estimativa baseada em pontos de função, no entanto, mesmo esta apresenta valores em linhas de código subvalorizados, uma vez que foi feita considerando que a aplicação do Lazer no Palm incluiria o módulo restaurantes e este módulo não chegou a ser desenvolvido. Caso o módulo dos Cinemas tivesse sido incluído (o que seria impossível neste prazo) as linhas de código no Palm provavelmente subiriam para o dobro.

Análise de Esforço e Duração

Uma vez que não era possível aumentar o número de pessoas no projecto e não existe possibilidade de adiamento na sua entrega, vimo-nos obrigados a aumentar o número de horas de trabalho mensal, aumentando o esforço despendido, de modo a conseguirmos apresentar um protótipo com qualidades demonstrativas do sistema dentro do prazo estabelecido.

Conclusão

Embora as estimativas não nos tenham fornecido valores correctos em relação às LC e esforço, tiveram grande utilidade uma vez que nos alertaram para uma complexidade do sistema que não era óbvia, na fase de análise de requisitos e desenho de sistema. Este “alerta” permitiu-nos determinar numa fase inicial da codificação, um conjunto de metas consideradas atingíveis e chegar a uma solução de compromisso com os objectivos pedagógicos da cadeira e os propósitos extra académicos do projecto.

11 Crítica ao Planeamento e análise de risco

Em relação às tarefas inicialmente apresentadas no planeamento efectuámos as seguintes alterações:

- Retirámos as tarefas relacionadas com o desenvolvimento da componente relativa aos Cinemas.
- Os métodos que comunicam com a Base de Dados são locais (criando ligações directamente à Base de Dados), em vez de serem remotos como foi inicialmente sugerido. Esta alteração deveu-se à incerteza das restrições de segurança impostas pelo servidor onde se localizariam os métodos remotos, que comunicariam com a BD Oracle através de ligações JDBC. Pelo que optámos por delegar esta decisão tecnológica para uma fase posterior, tendo o código sido gerado de modo a que possa ser reutilizado.
- Não foi criado o package com funções PL/SQL, uma vez que as queries executadas pela conduta devido à sua simplicidade não justificam o espaço gasto para alojá-las na Base de Dados uma vez que a melhoria de desempenho não seria significativa.

O planeamento que apresentámos na primeira fase foi revisto e apresentamos agora um planeamento que já não contém erros de super utilização de recursos humanos. No segundo planeamento revimos também os milestones tentando adequá-los ao tipo de projecto que realizamos.

Foi impossível seguir o calendário que propusemos devido a termos outras cadeiras que nos impunham datas de entrega faseada o que fez com que algumas vezes tivéssemos de parar um projecto e dedicarmos exclusivamente a outro durante uma semana ou mais.

Todas estas alterações ao planeamento se devem a riscos contemplados na nossa análise de risco, a verdade é que quase todos os riscos que tínhamos como muito prováveis ou prováveis acabaram por acontecer.

As condições especiais em que decorreu o nosso projecto (ser um projecto académico de último ano) levaram a que se tenham verificado mais que os 20 por cento dos riscos identificados como é habitual nos projectos de software. Nestas circunstâncias qualquer que fosse o nosso planeamento RMMM a sua execução seria sempre impossível.

Uma das situações que prevíamos na Análise de Risco era a da falta de conhecimento das tecnologias empregues, como monitorização tínhamos o fomento da investigação, como é óbvio esta investigação foi realizada mas em poucos dias e a par da investigação para mais três projectos o que a tornou ineficiente. No capítulo da Gestão tínhamos prevista a possibilidade de usar uma tecnologia alternativa, o problema é que se não temos tempo para investigar a fundo a tecnologia que estamos a empregar muito menos resta tempo para investigar tecnologia alternativas.

A situação que nos provocou maiores atrasos foi sem dúvida o tempo partilhado com os outros projectos, este risco foi por nós encontrado mas não fizemos um plano RMMM porque a única solução é desistirmos de alguma das cadeiras e essa opção nunca se pôs.

A nossa inexperiência levou-nos a fazer estimativas muito baixas como se pode constatar na “critica às estimativas”.

Por ultimo todas estas situações levaram a que se desse o risco que tínhamos colocado como muito provável que era o de termos pouco tempo para os testes de software, pensamos que foram realizados os suficientes para podermos garantir um desempenho aceitável, mas não os suficientes para um produto de qualidade final.

12 Conclusões e análise crítica do projecto

Neste projecto demos o primeiro grande passo no desenvolvimento do Guia Móvel do Lazer e terminámos o Guia do Lazer Personalizado.

O Guia Móvel do Lazer quando atingir a maturidade poderá vir a ser largamente difundido, dependendo no entanto, da popularização dos dispositivos Palm Pilot. Este sistema poderá vir a fazer concorrência a serviços semelhantes disponibilizados para telemóveis através de WAP, uma vez que apresenta uma alternativa mais económica e com maiores possibilidades de expansibilidade.

Este projecto apresenta viabilidade comercial, uma vez que poderá existir exploração publicitária em duas situações:

- Na página à qual o utilizador se deverá dirigir para autorizar a sincronização
- Na definição dos restaurantes a carregar para os dispositivos quando se dá a sincronização.

Quanto ao desenvolvimento do projecto, consideramos que embora tenhamos conseguido atingir metas importantes (e difíceis), ficaram por realizar etapas que inicialmente considerávamos atingíveis, este facto deveu-se principalmente à sobrecarga de trabalhos que sofremos.

Pela análise das Estimativas efectuadas e valores obtidos concluímos que as estimativas embora não forneçam dados exactos, são um forte indicador da dimensão dos projectos e deverão sempre ser alvo de um trabalho rigoroso nas análises preliminares dos projectos.

O Planeamento teve um papel muito importante no nosso projecto, uma vez que embora a calendarização não tenha sido rigorosamente cumprida, o cumprimento das datas de milestones permitiu-nos realizar diversas componentes do trabalho em paralelo e assim atingir um bom nível de produtividade.

Os Planos de Testes embora não tenham sido executados de forma exaustiva, permitiram-nos encontrar e resolver alguns problemas, principalmente nos testes de Interface Gráfica do Utilizador.

Quanto à criação dos manuais encontrámos uma certa dificuldade em realizá-los, uma vez que ao desenvolvermos uma aplicação, torna-se difícil olhá-la do ponto de vista do utilizador e explicar o seu funcionamento.

A nível pedagógico consideramos que a experiência adquirida ao desenvolvermos este projecto foi extremamente enriquecedora, uma vez que ao desenvolvermos um projecto diferente de todos os nossos colegas encontrámos dificuldades e situações inesperadas, as quais tivemos que resolver sozinhos, dando-nos assim uma maior auto confiança e sentido de responsabilidade. Esta situação deu-nos uma maior motivação para o desenvolvimento do projecto; , uma vez que se assemelha mais à vida “real” e não se trata de um projecto “controlado”, do qual já se sabe aproximadamente os resultados a obter.

13 Trabalho Futuro

- Desenvolvimento do módulo Cinemas.
- Adaptação do sistema às restrições de segurança do servidor da Base de Dados do Publico.
- Desenvolvimento da Interface Web que complementa o processo de sincronização (a integrar no site do Publico).

14 Bibliografia

- Pressman, Roger S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill, 4ª ed., 1997.
- Rumbaugh, James, Blaha, Michael, Premerlani, William, Eddy, Frederick and Lorenzen, William, *Object-Oriented Modeling and Design*, Prentice Hall, 1991.
- Albrecht, Allan J. and Gaffney, John E. Artigo "Software Function, Source Lines of Code, and Development Prediction: A Software Science Validation".
- Boehm, B, "*Software Engineering Economics*", Prentice-Hall, 1981, caps. 6,7,8.
- Developer's Guide for Java Based Conduits
- Documentação do CDKJ.
- Franck Allimant, Java® 2 SDK version 1.2.2 documentation
- António S. Ferreira, Acetatos das aulas práticas de PSI, 99/00
- Mário Silva, Acetatos das aulas teóricas de PSI, 99/00
- O'Reilly and Associates Inc, "Palm Programming: The Developers Guide"
- Documentação do Metroworks CodeWarrior Lite
- Palm.Com:The Development Zone, "Palm OS SDK Reference Guide"
- Palm.Com:The Development Zone, "Palm OS Programmer's Companion"
- Palm.Com:The Development Zone, "Palm OS 3.0 Tutorial"
- Grupos de Discussões, <news://news.massena.com/pilot.programmer>
- Grupos de Discussões, <news://news.massena.com/pilot/programmer.codewarrior>
- www.palmcentral.com