INSTITUTO SUPERIOR
DE TECNOLOGIAS
AVANÇADAS

Projeto Final

Gonçalo Cruz Nº45141
Diogo Silva Nº45148

CTESP- Desenvolvimento para Dispositivos Móveis

Sandro Ferreira
2018/2019

Projeto Final

CTESP- Desenvolvimento para Dispositivos Móveis

Índice

I. Introdução

A Aplicação que criámos faz parte do projeto Final da cadeira de Programação para Dispositivos Móveis III, foi-nos pedido para que a aplicação tivesse um mecanismo de autenticação, um currículo do aluno que desse para consultar e editar, fazer a consulta do horário, consulta de notas, consulta de propinas e que desse para fazer a consulta de um calendário de exames.

Utilizamos um dispositivo de versionamento bastante conhecido entre programadores, o github onde criamos um repositório com o nome “Projeto Xamarin” e para onde fazíamos “Commits” quando alguma funcionalidade estava feita, este também nos ajudou bastante a organizar o projeto, pois assim conseguimos saber o que cada um estava a fazer no projeto e em que funcionalidade estava a trabalhar.

Esta aplicação demorou-nos algum tempo a fazer devido à sua complexidade, tivemos de utilizar alguns recursos externos para que esta pudesse ser concluída.

II. Funcionalidades

Este projeto é constituído por sete funcionalidades que são: Login e Registar, currículo que desse para consultar e editar, fazer a consulta do horário, consulta de notas, consulta de propinas e que desse também para realizar a consulta do calendário de exames.

O Login e Registar são duas funcionalidades simples, o Login serve para caso já se tenha registado, coloca-se o user e a passe e têm-se acesso á aplicação, o Registar é para que o user crie uma conta e depois da conta feita possa entrar na aplicação.

O Curriculo mostra o email, telefone e morada do utilizador,

O horário é as aulas que cada turma tem durante a semana.

As notas é a classificação dos alunos de cada turma por cadeira.

As propinas é um x de dinheiro que o aluno tem de pagar todos os meses, na aplicação aparece se está pago ou não x mês e opção do aluno confirmar se pagou ou não.

O calendário de exames é os exames que uma turma tem marcados para fazer.

III. Tecnologias Usadas

Para a realização deste projeto, foi utilizado para o desenvolvimento da aplicação o Xamarin.Forms, que faz parte do Visual Studio. Foi utilizada a linguagem de programação C# para programar no Xamarin.Forms.

Foi utilizado uma base de dados mysql para guardar,criar e atualizar os dados que serão utilizados pela aplicação. Esta base de dados é guardada no web host “000webhost” para a aplicação conseguir aceder à base de dados online.

É utilizado phps para realizar a ligação entre a aplicação e a base de dados.

Foi utilizado o github que é uma plataforma de hospedagem de ficheiros utilizando mecanismos de versionamento .git e serviu para guardar todos os ficheiros necessários para a realização do projeto(projeto,phps,etc.).

IV. Conceitos Usados

Para o Xamarin.Forms, os conceitos utilizados para os layouts(.xaml) foram:

- Os Stack Layouts que serviram para dispor os elementos dentro do stack de uma forma linear tanto vertical,como horizontal (utilizados na maioria da aplicação).
- O MasterDetailPage que serviu para criar um menu com barra lateral, <MasterDetailPage.Master>, onde encontram-se botões para alterar o <MasterDetailPage.Detail> (Página), que serve como um recipiente para as páginas associadas aos botões (Carregar no botão currículo altera o <MasterDetailPage.Detail> para a página Currículo) (utilizado no menu).
- O Grid para criar uma grelha onde dispor os elementos por colunas e linhas, utilizando <RowDefinition> para definir as linhas e <ColumnDefinition> para definir as colunas, e colocando os atributos Grid.Row e Grid.Column nos elementos para determinar em que linha e coluna esse elemento será colocado (utilizado no horário).

- O ListView que serviu para criar uma lista de entradas da base de dados para ficarem alinhados da mesma forma(Utilizando {Binding} nos elementos para ligar as colunas com o mesmo nome de cada entrada).Foi também colocado dentro dos elementos das listas eventos para atualizar a lista Refreshing e para quando o utilizador carrega num item da lista ItemSelected.

Os conceitos utilizados para o código(.cs) foram:

- await Navigation.PushAsync() e PushModalAsync() para abrir e mover para novas páginas na aplicação (os métodos que utilizam estes métodos têm a propriedade async incluído para serem assíncronos).
- Classes DBConnections para fazer a conexão entre a aplicação e a base de dados.
- Utilização de métodos Task<string> dentro das DBConnections para realizar a tarefa específica com a base de dados(Pull,Update,Get) em que cada um é utilizado uma String URL com o URL da página(php) a ser utilizado para esta conexão e o envio de uma ou mais strings (por vezes não é enviado nada) para a base de dados receber essas strings.

- Classes em que as variáveis delas são iguais às colunas da tabela correspondente, para facilitar o recebimento dos dados da base de dado dessas tabelas(As variáveis têm propriedade set e get para poderem ser facilmente alterados e recebidos.
- Utilização de uma class Hash para codificar a palavra-passe escrita no registo(para não ficar a palavra passe descodificada na base de dados).
- await DisplayAlert() para exibir ao utilizador Caixas de alerta quando algo não corre bem(deu erro).
- Utilização de variáveis státicas, encontradas na classe Variáveis, para serem utilizadas pela aplicação toda(identificar qual o aluno atualmente ligado, guardar todos os valores das tabelas, etc.).

- Utilização de JsonConvert.DeserializeObject<Class>(result) para desfazer o JSON recebido pela base de dados(a base de dados envia JSON para comunicar com a aplicação), em que Class é a Class do tipo da tabela que foi enviada pela base de dados(se a base de dados envia um JSON da tabela aluno, é utilizado a class Aluno) e o result é uma string que espera pelo resultado de uma Task<string> que vai buscar o JSON à base de dados (string result = await resultTask) .
- Utilização de método OnAppearing() para executar código antes da página aparecer ao utilizador.
- Utilização de identificadores x:Name nos elementos do .xaml para serem modificados por código .cs (alterar texto, obter texto, etc.).
- Utilização da Variável Detail no MenuPage para alterar a página da aplicação, e mantendo a barra lateral acessivel para mudar para outra página.
- Utilização de um array coluna com arrays para cada coluna no horário para obter a localização de cada linha e coluna na Grid (colunas[coluna][linha]) para ser colocado numa linha e coluna especifica o texto correspondente.

- Criação de uma Variável ObservableCollection<> nas páginas com ListViews para converter elementos de um array string de uma classe para ObservableCollection<> para poderem ser utilizados pela ListView(alguns elementos são selecionados dependente se têm um valor de variável requerido, por exemplo na pagina exame é requerido que os elementos selecionados têm o valor turmaId igual á variavel static turmaId).
- Utilização do evento Refreshing da ListView para executar o código necessário para atualizar a ListView
- Utilização do evento ItemSelected da ListView na página Propinas para exibir uma ActionSheet em que contém 2 botões (Sim ou Não) em que é executado 2 códigos diferentes dependendo de qual botão é selecionado(o sim manda atualizar o valor pagoN na base de dados com “Propina Paga” e não manda atualizar com “Propina não Paga”).

Os conceitos utilizados na criação dos phps foram:

- Um connection.php para abrir a ligação do php com a base de dados.

- Um var.php onde vai receber os Posts da aplicação(realizado de uma forma em que se o php não recebe uma certa variável da aplicação, essa variável é ignorada).
- utilização de uma variável statement que vai declarar o que o php irá realizar(Select,Update,Insert) e definir quais são as variáveis que vai utilizar (no caso de se a declaração tiver o nome da variável seguido por um ponto de interrogação username = ?).
- Com o mysqli_stmt_bind_result, é juntado a cada variável o correspondente valor na coluna com o mesmo nome da variável.
- Utilização de while(mysqli_stmt_fetch($statement)) para criar um Array com as variáveis .
- Utilização de echo json_encode para criar um JSON a partir do array e enviar o JSON para a aplicação.
- Os phps com pull e o nome da tabela ou o php login servem para retornar os elementos da tabela com o id que a aplicação fornece.

- Os phps com pullRow e o nome da tabela servem para retornar o número de linhas da tabela correspondente (isto serve para criar um array na aplicação com o número destas linhas).
- O php register serve para criar um aluno consoante os valores recebidos pela aplicação (name,username,password).
- O php com postCurriculo serve para criar um curriculo vazio com o id do aluno associado para o aluno conseguir colocar as informações lá a partir do php updateCurriculo (telemovel,morada,email).
- O php updatePropinas serve para alterar o valor da coluna pagoN da tabela Propinas, tendo em conta o seu id.

V. Como replicar o Projeto?

Para replicar este projeto é bastante simples apenas é preciso fazer algumas coisas que são:

1) Criar uma conta no 00webhost.
2) Fazer clone ao repositório
3) Colocar os ficheiros PHP no file Manager do 00Webhost
4) Importar a base de dados para o 00Webhost
5) Alterar os links que estão no projeto pelos novos links de cada PHP
6) Instalar a APK no telemovel.

IV. Conclusão

Para concluir este relatório, este projeto foi o fim de uma cadeira em que aprendemos como criar aplicações em Xamarin com o professor Sandro Ferreira. Com este projeto desenvolvemos a capacidade de ir á procura de soluções para os diversos problemas que foram aparecendo ao longo de todo o projeto. Aprendemos a programar em C# e a criar layouts em XAML.