

Embarcadero Conference

Um único esforço, uma única base de código, múltiplas
plataformas, múltiplos dispositivos



MAICO LOPES DAL RI

Usando Notificações em Dispositivos Móveis



AGENDA

- O que são notificações
- Acessando os serviços de notificações
- A central de notificações (NotificationCenter)
- Manipulando Notificações
- Configurando e Vinculando ações a uma notificação
- Agendando Notificações
- Cancelando uma notificação agendada
- Notificações Remotas (Push Notifications)



O que são notificações

O sistema de notificação permite que o seu aplicativo mantenha o usuário informado sobre eventos, tais como novas mensagens de chat ou um evento do calendário.

Pense em notificações como um canal de informações que alerta o usuário para eventos importantes como eles acontecem ou um registro que narra eventos enquanto o usuário não está prestando atenção.



Acessando os serviços de notificações L SEP

Para acessar os serviços de notificações em Delphi é necessário fazer uso de 2 units

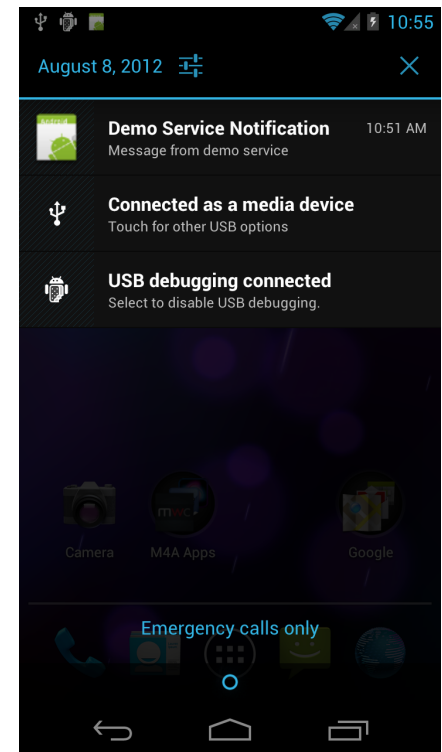
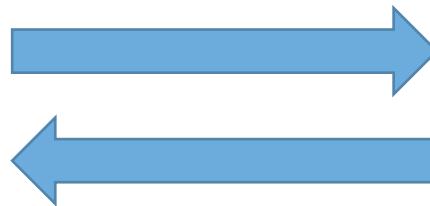
FMX.Platform – XE4 introduziu um recurso chamado “Platform Services” que é útil para retornar se a plataforma que irá executar o aplicativo realmente suporta um determinado serviço

FMX.Notification – Mapeamento de classes da plataforma corrente para utilização do Delphi

A central de notificações (NotificationCenter) LSEP

Classe que permite utilizar os recursos de notificação do dispositivo móvel

NotificationCenter





Criando Notificações

```
procedure TForm1.Notificar;  
var  
    MyNotification: TNotification;  
Begin
```

```
1.  MyNotification := NotificationCenter.CreateNotification;  
2.  try  
3.      MyNotification.AlertBody := 'Exemplo de notificação';  
4.      NotificationCenter.PresentNotification(MyNotification);  
5.  finally  
6.      MyNotification.DisposeOf;  
7.  end;
```

```
end;
```





Vinculando ações a uma notificação

A classe `TNotificationCenter` fornece um manipulador de eventos *onReceiveLocalNotification* que lhe permite escrever uma resposta quando o usuário clica na mensagem de notificação no centro da notificação. Para escrever a resposta, clique duas vezes no componente `TNotificationCenter` no Formulário de FireMonkey Mobile Designer, e em seguida, implementar o manipulador de eventos `OnReceiveLocalNotification`.



O seguinte trecho de código implementa uma resposta para mostrar uma caixa de mensagem que diz qual notificação foi acessada.

```
procedure TForm1.NotificationCenterReceiveLocalNotification(Sender:
TObject; ANotification: TNotification);
begin
    ShowMessage('Notificação [' + ANotification.Name + '] acionada.' );
end;
```


Agendando Notificações ^LSEP



```
procedure TForm1.ActionNotification;
var
    MyNotification: TNotification;
begin
    MyNotification := NotificationCenter.CreateNotification;
    try
        MyNotification.Name := 'Minha Notificação';
        MyNotification.AlertBody := 'Notificação Agendada';
        MyNotification.FireDate := Now + EncodeTime(0, 0, 20, 0);
        NotificationCenter.ScheduleNotification(MyNotification);
    finally
        MyNotification.DisposeOf;
    end;
```

Cancelando uma notificação agendada ^LSEP

```
procedure TForm1.CancelNotification;  
begin  
    NotificationCenter.CancelNotification('Minha Notificação');  
end;
```



Notificações Remotas (Push Notifications) L SEP

Sistema reativo “Para quê manter um programa de comunicações sempre em execução, à espera de uma hipotética mensagem que poderá nunca chegar?”

Push no Android e no iOS (Protocolos)

iOS: Apple Push Notification APN

Parse

Kinvey

Android: Google Cloud Messaging GCM

Kinvey

Obrigado a Todos pela
Atenção

www.aquasoft.com.br

maicodalri@aquasoft.com.br

<http://www.embarcadero.com/br/products/delphi>

“Deus não joga dados”

- Albert Einstein