

PARALELISMO COM DELPHI NO DESKTOP E MOBILE

Marcelo Varela

Embarcadero

Conference

Marcelo Varela - mini currículo

Embarcadero MVP (Most Valuable Professional)

Principais Certificações

- Delphi Developer Certification 7/2005/2006/2007/XE
- Master Delphi Developer Certification
- Delphi Instructor Certification

Palestras e eventos

- Delphi Conference Brasil 2009, 2010, 2011, 2012, 2013, 2014
- BonCon 2005, 2006 e 2007

Atividade profissional

- Gerente de projetos de Lógica Treinamentos - Natal/RN
- Master Trainer da Lógica Treinamentos
- 20 anos em experiência em desenvolvimento de software

Atividade acadêmica

- Professor de Sistemas de Informação do IFRN
- Mestre em Engenharia de Software na UFRN/IMD
- Pós-Graduado em Desenvolvimento de Sistemas Web pela UnP
- Bacharel e Sistemas de Informação

Artigos

- ActiveDelphi, Linha de Código e iMasters.





Agenda

Parallel Programming Library

- Task
- Future
- For

Dicas

- Conclusão

Parallel Programming Library

Embarcadero

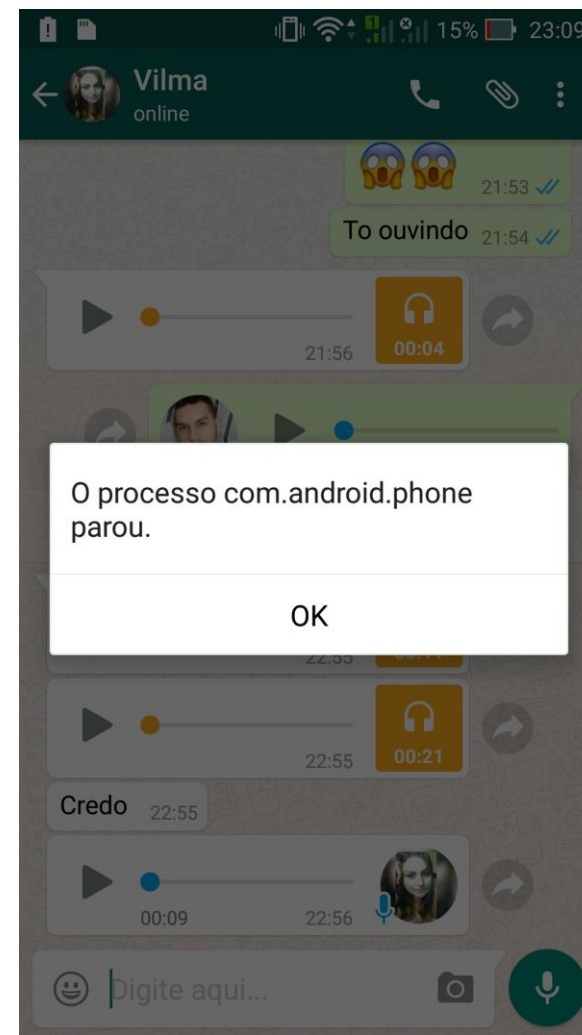
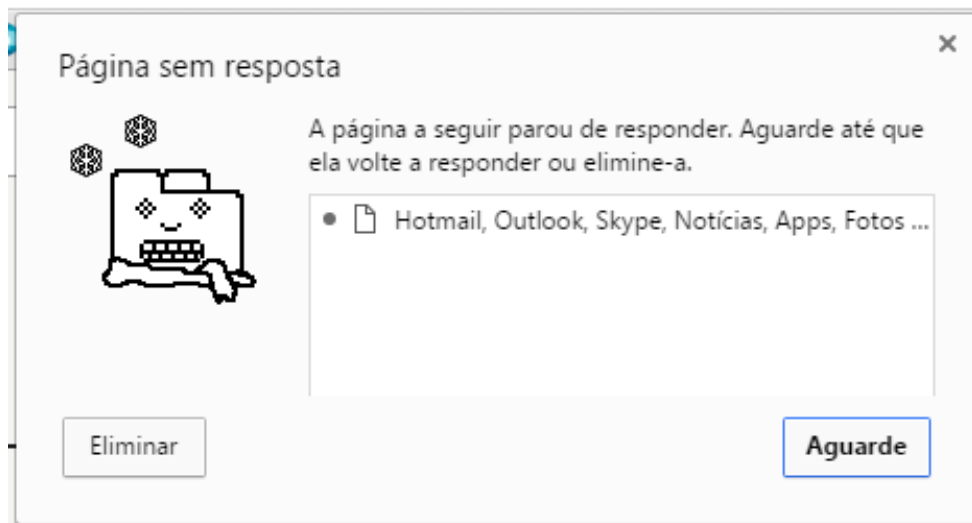
Conference



UX – User eXperience

- UX — User Experience (Experiência do Usuário) é a competência responsável por projetar experiências de uso encantadoras para fidelizar e conquistar clientes baseada em três pilares:
 - Utilidade — O quão útil é o serviço para o cliente; quanto é melhor fazer as coisas usando o serviço e não alguma alternativa.
 - Facilidade de uso — O quão fácil e rápido é usar o serviço e resolver o que for preciso usando ele e não uma alternativa.
 - Prazer — O quão prazeroso (divertido, interessante, recompensador, etc.) é usar o serviço e não as alternativas.

Aplicação travada





Por que escrever código assíncrono?

- Operações assíncronas no nosso código são importantes quando desejamos executar tarefas que podem ser demoradas ou de segundo plano, sem que afetem a experiência de interação do usuário com a interface do nosso software.
- Operações assíncronas nos possibilitam ter uma interface gráfica responsiva, que não fica bloqueada esperando alguma operação terminar, ou impossibilitando que o usuário continue seu trabalho.
- Interfaces que travam, podem aparentar lentidão no processo, cancelamento desnecessário ou mesmo retrabalho do usuário.



Parallel Programming Library

- É uma nova biblioteca, disponibilizada a partir da versão atual, XE7, que torna o trabalho do desenvolvedor mais simples no que se referente a programação paralela
- Ela promove uma facilitação na criação e gerenciamento de novos segmentos dentro do seu processo principal. É como trabalhar com Thread, só que mais fácil e poderoso
- Multiplataforma
- `System.Threading`
- Task, Future e For (loop)



Task

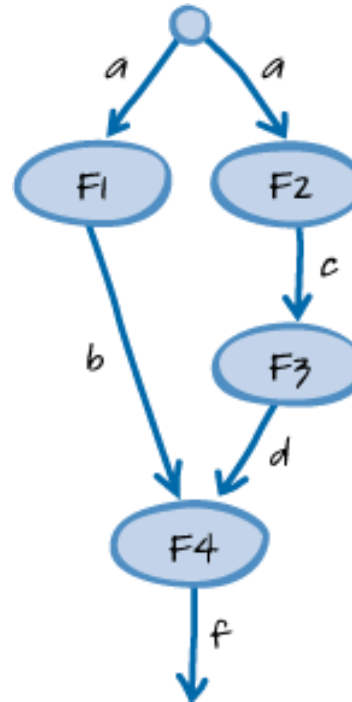
- Uma Task (tarefa) funciona como *procedure* que roda em paralelo, ou seja, uma rotina sem retorno.
- Uma Task não inicia imediatamente após a chamada do método Start, elas são enfileiradas e iniciadas quando há disponibilidade na CPU.

Future

- Uma Future (tarefa futura) funciona como *function* que roda em paralelo, ou seja, uma rotina com retorno.

- Veja um exemplo:

- $a := 10;$
- $b := F1(a);$
- $c := F2(a);$
- $d := F3(c);$
- $f := F4(b, d);$
- $Result := f;$





Task/Future

- Tasks também podem ser encadeadas com outras Tasks e aguardar o final de uma tarefa ou de todas. Para isso usamos `WaitForAny` e `WaitForAll`.



For

- Um For, usando a PPL, divide um laço FOR em pedaços que rodam em paralelo.
- A chamada de um Parallel.For não é assíncrona, por isso, se desejar evitar o efeito *freeze* de tela, use em conjunto com uma Task.



Dicas

- Cuidado com a concorrência de recursos
- Preocupe-se em deixar sua aplicação responsiva
- Use sempre o Synchronize/Queue quando for acessar/modificar componentes



Conclusão

- A PPL é uma biblioteca pronta para você usar os recursos de paralelismo e multicore.
- Você não tem mais desculpas para criar aquelas Apps com a tela travando!

OBRIGADO



Marcelo Varela

marcelo@logicaetreinamentos.com.br

@marcelologica

Embarcadero

Conference





Embarcadero Conference