



Arquitetura de aplicações multiplataforma para dispositivos móveis

Prof. Ilo Rivero
(ilo@pucminas.br)

O que é Arquitetura de Software?

- Os fatores que compreendem o projeto no nível arquitetural e estão diretamente relacionados com a **organização do sistema** e, portanto, afetam os atributos de qualidade (também chamados de requisitos não funcionais) como **desempenho, portabilidade, confiabilidade, disponibilidade**, entre outros

O que é Arquitetura de Software?

- O entendimento das arquiteturas permite aos engenheiros tomarem decisões sobre alternativas de projeto
- Uma especificação arquitetural é essencial para analisar e descrever propriedades de um sistema complexo, permitindo o engenheiro ter uma visão geral completa do sistema

O que é Arquitetura de Software?

- O conhecimento de notações para descrever arquiteturas permite engenheiros comunicarem novos projetos e decisões arquiteturais tomadas a outros membros da equipe
- A arquitetura define a estrutura do sistema, de modo consistente para implementações, já que está diretamente relacionada aos atributos de qualidade como confiabilidade e desempenho

O que é Arquitetura de Software?

- Dividir a arquitetura em várias camadas (4 ou 5) pode degradar o desempenho do sistema se houver um elevado grau de acoplamento entre os componentes

Arquitetura para Dispositivos Móveis

- No caso de dispositivos móveis o uso de múltiplas camadas que facilita a manutenção contribui para degradar o desempenho do sistema
- Uma tática tem sido usada, é reduzir o nível de acoplamento. Se adotarmos uma redução no nível de acoplamento dos componentes, teremos menor necessidade de comunicação entre si, o que resulta num melhor desempenho

Arquitetura para Dispositivos Móveis

- Observações importantes para definição do estilo arquitetural a ser usado em dispositivos móveis:
- **Identificação de componentes:** autenticação, cadastro etc
- **Identificação de mecanismos de interação:** troca de mensagem entre componentes

Estilos de Arquitetura de Software

- **Invocação implícita** – diferentemente do estilo baseado em objetos, no qual um componente invoca outro diretamente por meio da passagem de mensagens, a invocação implícita requer que componentes interessados em receber ou divulgar eventos se registrem para receber ou enviar. Um exemplo de sistema empregando mensagens são listas de notícias e fórum

Estilos de Arquitetura de Software

- **Eventos** – trata-se de um estilo no qual os componentes podem ser objetos ou processos e a interface define os eventos de entrada e saída permitidos
- Conectores são implementados através do '*binding*' evento-procedimento. Assim, eventos são registrados junto com eventos e os componentes interagem entre si através do envio de eventos

Estilos de Arquitetura de Software

- Qual é a melhor para dispositivos móveis?
- Não existe uma única, dependo do modelo de desenvolvimento escolhido : Web app, app híbrida ou app nativa.
- A mais comum é a de camadas.
- Para melhorar a questão do performance, usa-se o binding e filtros de notificação.

Desafios

- Não se deve pensar em apenas replicar para smartphones e tablets, os sistemas escritos para PC
- Sistemas inovadores, que explorem os recursos dos equipamentos :
- Câmeras; Microfones; Giroscópios; Acelerômetros; GPS e diversos outros.

Desafios

- A multiplicidade de sensores destes equipamentos nos permite criar aplicativos que capturem e processem dados oriundos das mais diversas fontes
- Isto por si só é um desafio, pois os diversos fabricantes configuram estes recursos de formas diferentes. Além disso, estes recursos podem gerar volumes de dados significativos como os vídeos

Desafios

- Também as interfaces via reconhecimento de voz e som demandam novas maneiras de pensar os aplicativos
- Os programas têm que explorar a capacidade do próprio dispositivo, mas também precisam da imensa capacidade computacional que estará rodando em background, na nuvem

Desafio

- Um exemplo: um app que grave um vídeo e o envie para processamento na nuvem para que seja feito um reconhecimento de imagens que identifique determinado indivíduo ou produto neste vídeo e forneça características sobre ele

Desafio

- Outra análise a ser feita para as aplicações móveis é o grau de computação que deverá ser feita offline, quando o acesso à rede não for possível
- Pode acontecer do app ter que operar em determinados momentos em locais onde o sinal seja muito fraco ou mesmo sem possibilidade de conexão à Internet. Na prática, podemos pensar em um modelo mais flexível que o cliente-servidor

Desafio

- A grande capacidade computacional de um smartphone ou tablet permite que ele seja, em determinados momentos, cliente de servidores na nuvem (client-cloud)
- A sincronização com a nuvem deverá ser feita quando a conexão se restabelecer, de forma automática e transparente para o usuário do app

Desafio

- **Outro desafio:** criar arquiteturas para apps híbridas, adotando HTML5 e código nativo para Android, iOS e Windows
- Isto implica em conhecimentos não apenas de HTML5, mas também da tecnologia específica dos demais ambientes móveis, como dos diversos meios de comunicação destes dispositivos com a nuvem.

Desafio

- Por exemplo, se o acesso for a partir da parcela nativa do app, deverá ser feita via API específica da plataforma
- Se for a partir do código HTML5, será via Ajax ou Web Sockets. Enfim, são decisões arquitetônicas que deverão ser tomadas antes de escrever o código

Resumo dos Desafios

- Conectividade:
 - Largura de Banda Limitada
 - Conectividade Ocasional
 - Protocolos eficientes
 - Caching local de informações
 - Sincronização ocasional

Resumo dos Desafios

- Processamento:
 - Algoritmos Eficientes
 - Processamento remoto
 - APIs otimizadas para minimizar consumo de bateria
 - Quantidade limitada de Memória
 - Baixo poder de processamento
 - Fonte de energia finita (bateria)
 - Execute processamentos longos em background

Resumo dos Desafios

- Usabilidade Desenvolvimento:
 - Espaço de display
 - Prototipe a interface
 - Minimize a entrada de dados

Soluções para o desenvolvimento

- O uso de aplicativos móveis tem apresentado altas taxas de crescimento ao redor do mundo todo. Isso porque as pessoas estão cada vez mais conectadas aos seus tablets e smartphones, utilizando os aplicativos seja para se comunicar, para jogar ou para comprar produtos

Soluções para o desenvolvimento

- Tendo em vista esse cenário, as empresas têm percebido a importância de desenvolver aplicativos móveis caso queiram estar presentes na vida das pessoas e atender às necessidades de seus clientes
- Mas como escolher a melhor solução?

Plataformas

- No mercado existe uma explosão no número de smartphones no mercado e conseqüentemente começa a haver uma necessidade de apoiar todo um conjunto de plataformas populares tais como IOS, Android e etc

Vantagens

- Diversidade de público
- Atingir o maior número de usuários
- Independência de dispositivos
- Fontes de receita diversificada
- Estratégia de venda diversificada

Problema no desenvolvimento

- Falta de requisitos mínimos por dispositivos
- Dificuldade na elaboração das interfaces
- SDKs diferentes
- Dificuldade de encontrar profissionais capacitados nas tecnologias
- Custo de desenvolvimento
- Políticas diferentes em cada loja

Desenvolvimento

- De modo a compreendermos a arquitetura multiplataformas temos que primeiro compreender os tipos de estratégias usadas no desenvolvimento de aplicações para dispositivos móveis
 - Aplicação Nativa
 - Aplicação Web
 - Aplicação Híbrida

Aplicações Nativas

- São aplicações desenvolvidas no SDK da plataforma
- Vantagens:
 - Alto desempenho
 - Permite maior integração com o S.O.
 - Interface dentro do padrão do S.O.

Aplicações Nativas

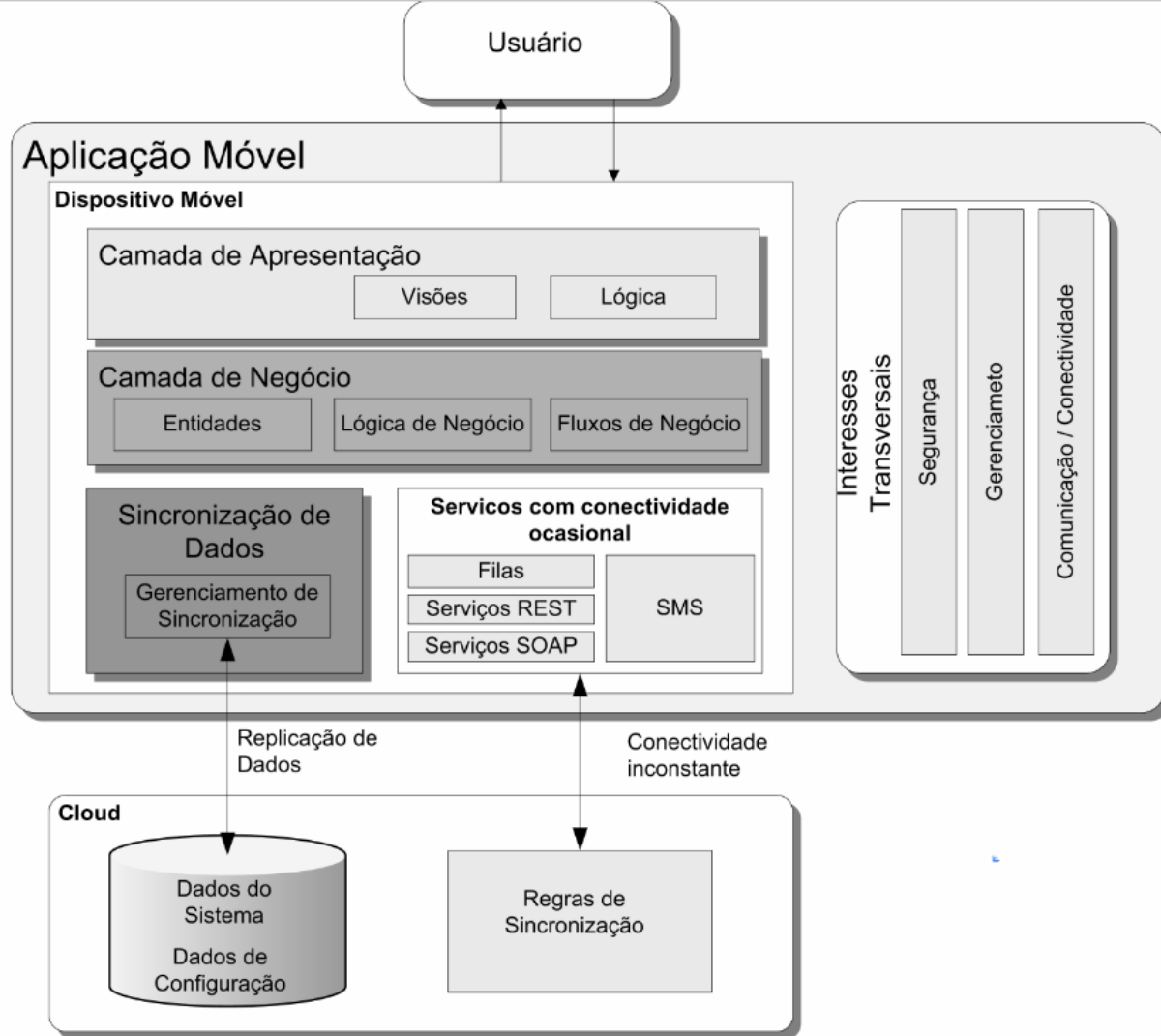
- Vantagens:
 - Comunicação entre aplicações do S.O.
 - Publicação na loja
 - Captura de notificações
 - Funciona de modo off-line
 - Acesso aos dispositivos

Aplicações Nativas

- Desvantagens:
 - Código complexo de ser migrado
 - Alto custo de desenvolvimento
 - Tempo de desenvolvimento largo
 - Manutenção com pouco reaproveitamento de código

Exemplo de Aplicativos Nativos

- Redes sociais
- Jogos
- Aplicativos que usam componentes dos smartphone (Câmera, GPS, Mapas, Lista de contato, Microfone e etc)



Web App

- São sites que possuem uma versão para dispositivos móveis
- O “aplicativo” é, na verdade, um site com um layout otimizado para dispositivos com tela pequena e sem uso de mouse para interagir
- Com o uso de JavaScript, um site desse tipo pode ter tanta interatividade quanto um aplicativo nativo

Web App

- Vantagens:
 - Multiplataforma
 - Tecnologias já conhecidas
 - Fácil manutenção para todos os sistemas
 - Custo de produção menor
 - Tempo de produção baixo

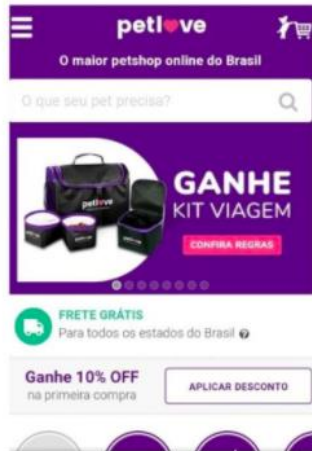
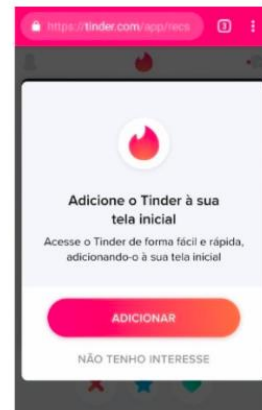
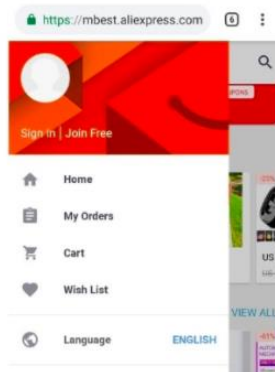
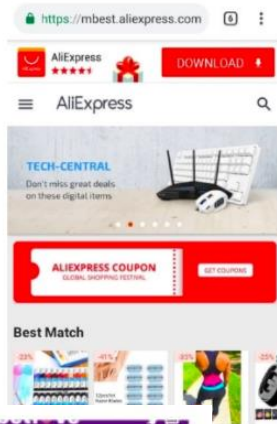
Web App

- Desvantagens:
 - Baixa integração com o Sistema Operacional
 - Interface fora do padrão da plataforma
 - Desempenho baixo
 - Arelada as limitações dos navegadores móveis
 - Alto consumo de dados

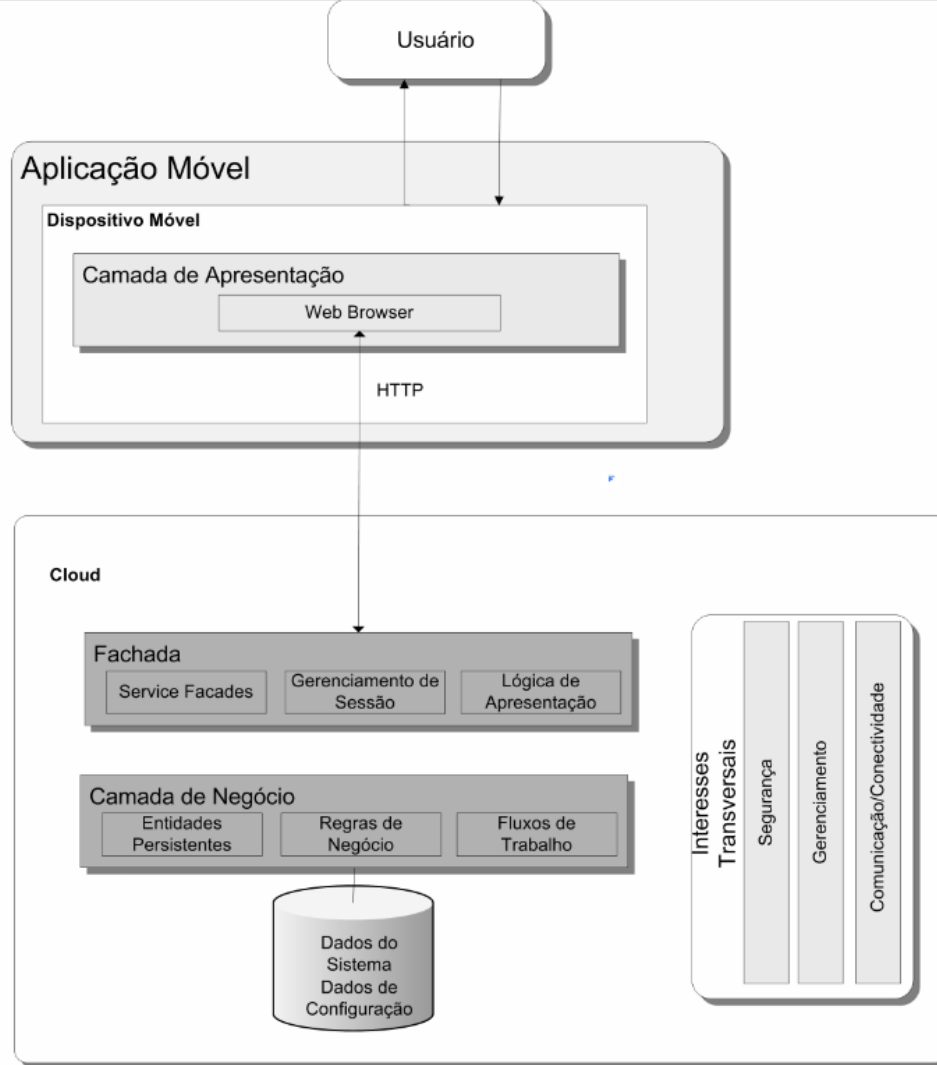
Web App

- Desvantagens:
 - Não funciona sem conexão de dados
 - Não está presente nas lojas
 - Não acessa os periféricos dos dispositivos

Web App - Exemplos



PUC Minas



Aplicação Híbrida

- Em uma arquitetura multiplataforma são utilizados híbridos, a aplicação pode ser desenvolvida usando tecnologias Web tais como HTML5, Javascript e CSS mas que rodam dentro da “Shell” nativo da plataforma Mobile, ou podem ser gerados códigos nativos traduzidos pela ferramenta.

Aplicação Híbrida

- Graças à sofisticação cada vez maior de ferramentas multiplataforma, o desempenho está melhorado drasticamente

Aplicação Híbrida

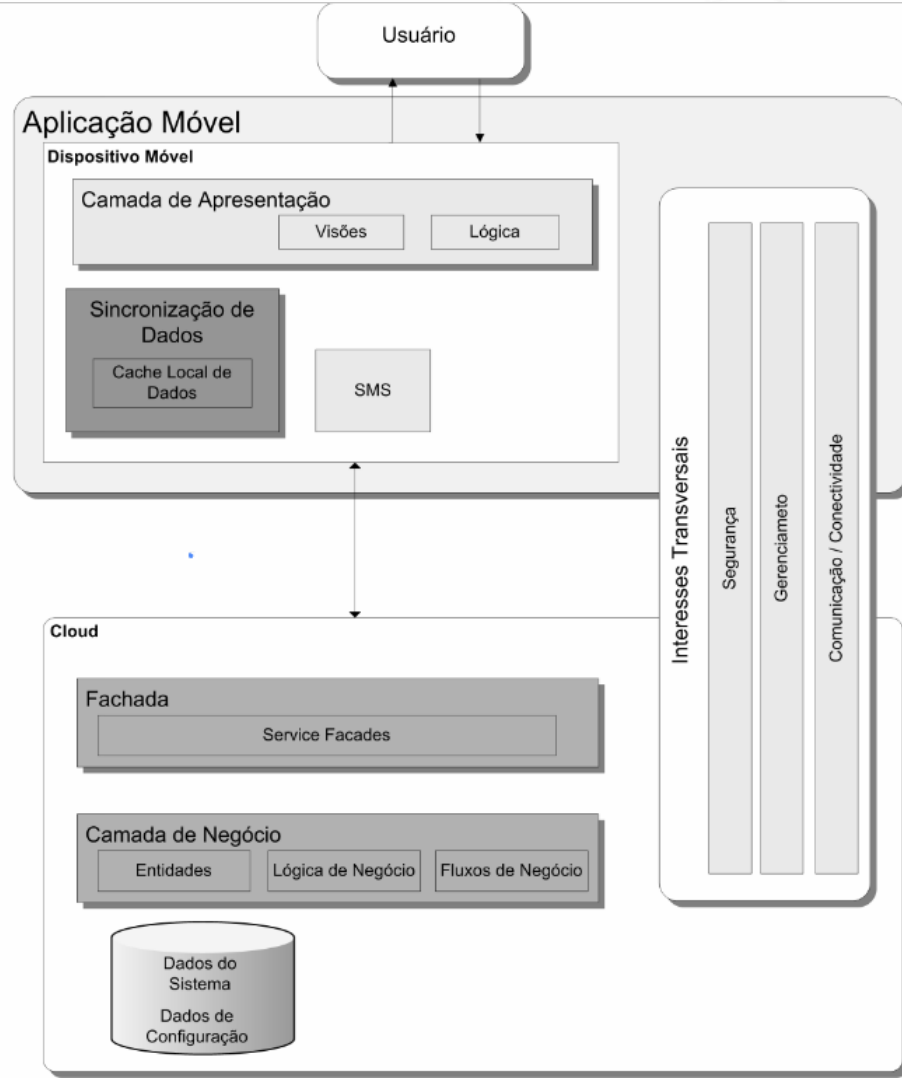
- Vantagens:
 - Permite comunicação com hardware via plugins ou tradução para código nativo
 - O código pode ser utilizado para várias plataformas
 - Usa somente uma tecnologia para exportar para vários S.O.s

Aplicação Híbrida

- Vantagens:
 - Integração com a UI
 - Funciona de modo off-line
 - Aumenta o número de usuários a um custo baixo
 - Aproveita o processamento do dispositivo do cliente

Aplicação Híbrida

- Desvantagem:
 - Não permite alta integração com o S.O. para acesso a aplicações
 - Os códigos são padronizados para as funções que existem em todas as plataformas
 - Possui uma performance média



1st
Gen

Platform Proprietary



Objective-C / Swift



Android Java



Windows .NET

2nd
Gen

Hybrid HTML & JavaScript Frameworks



PhoneGap



CORDOVA



ionic

3rd
Gen

Cross-Platform Native



Xamarin



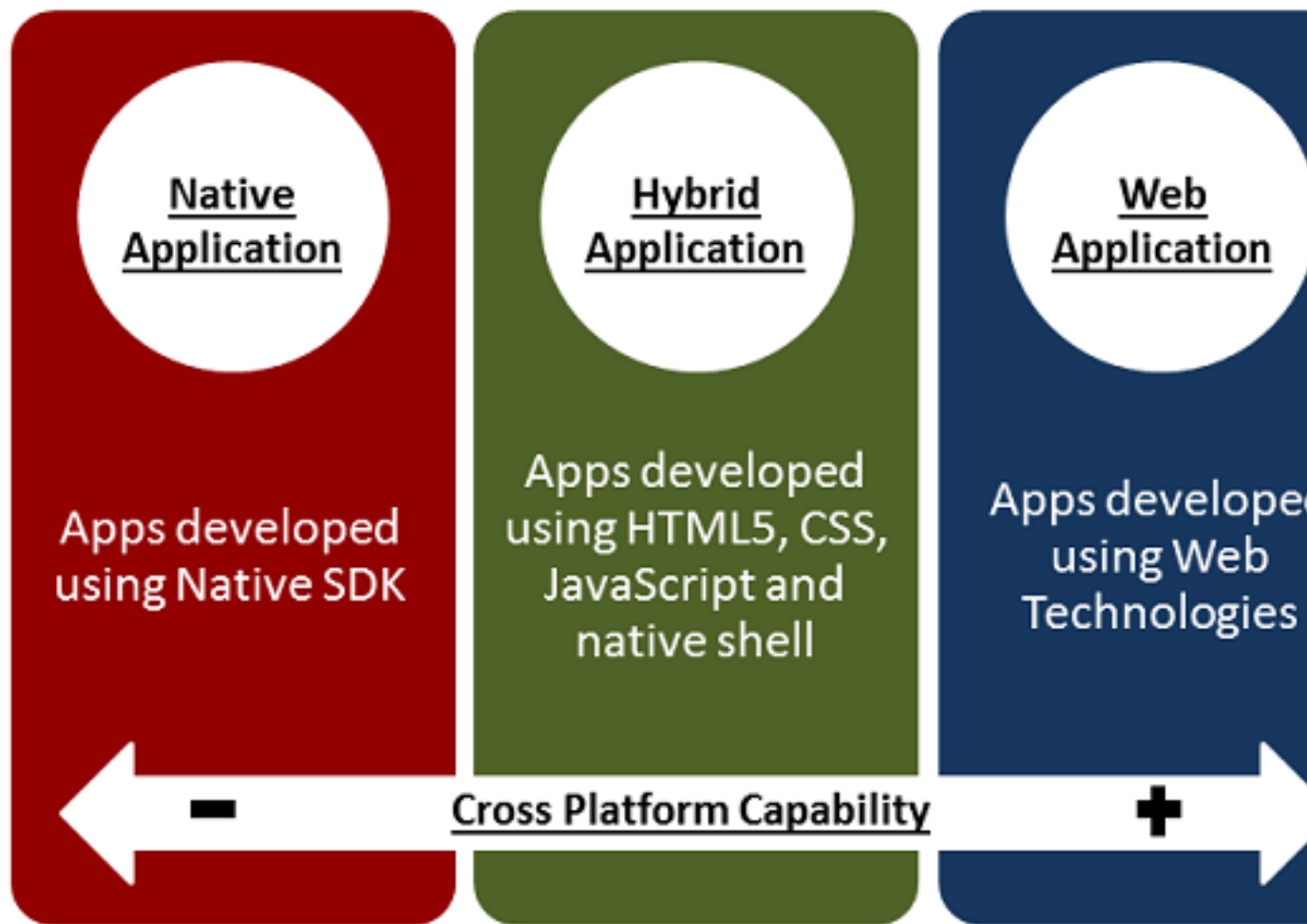
React Native

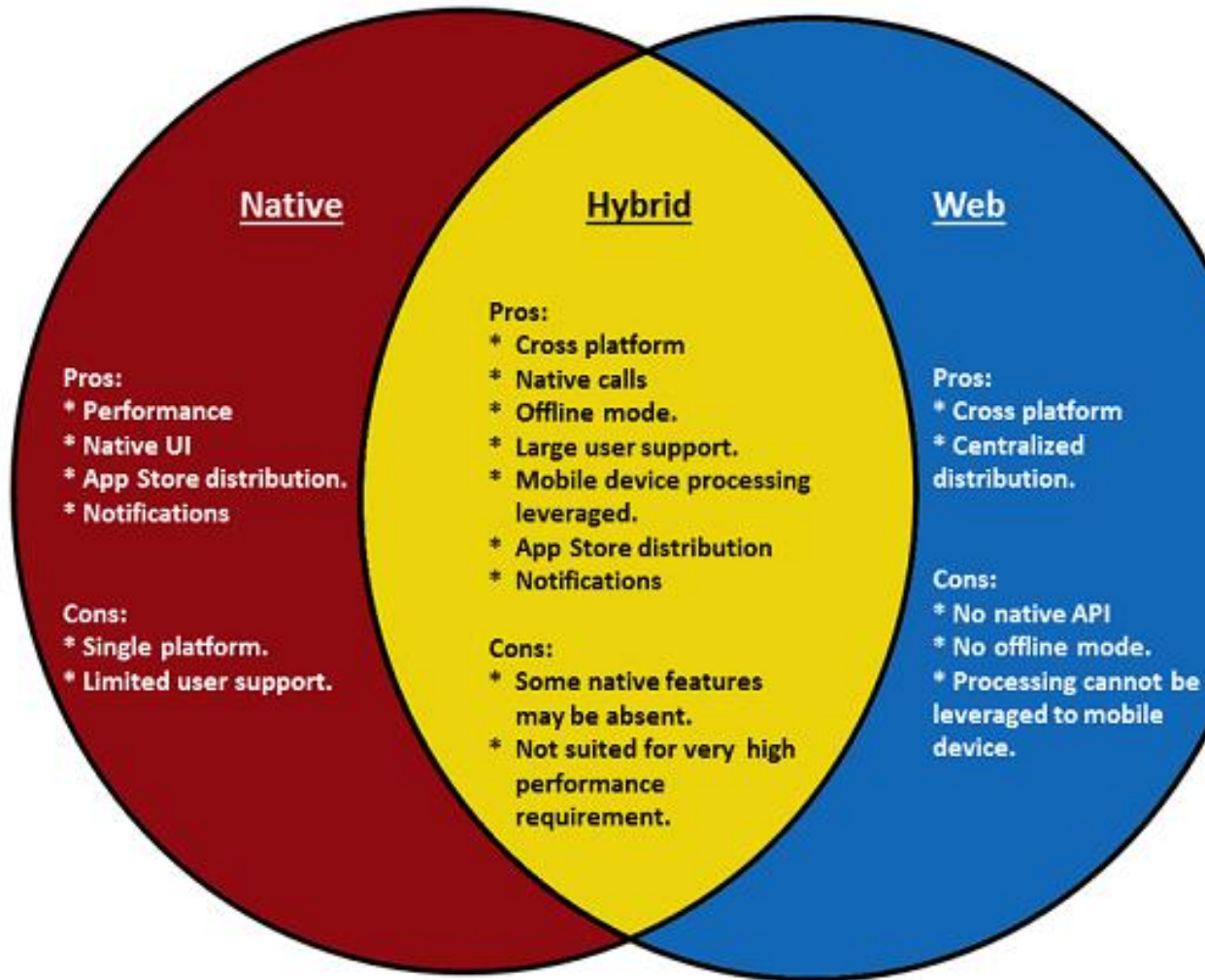


NativeScript

Cross Platform

- Uma variação são os aplicativos chamados cross-platform, que são aqueles que você desenvolve em uma linguagem específica e um gerador cria versões nativas para cada uma das plataformas que são o seu target.





Referências

ALVES, Rafael Félix; SILVA, Everaldo Leme da (org.). **Arquitetura para computação móvel**. 2. ed. São Paulo: Pearson, 2020. E-book. ISBN 9786550110581. (Capítulo 2)

Material de aula do prof. Arthur Mol.