**Desenvolvimento MULTIPLATAFORMA**

# Quando se tem um software que pode ser usado em sistemas diferentes ou esse software e capaz de comportar arquivos provenientes de diferentes sistemas , fazendo com que esses arquivos funcionem em um mesmo ambiente ou o software em ambientes diferentes com as mesmas funcionalidades, o chamamos de multiplataforma. Assim acontece com o desenvolvimento que e voltado para este propósito. Diz-se multiplataforma um programa ou sistema que roda em mais de uma plataforma.

# Para um programa ser desenvolvido com vista ao suporte multiplataforma, um conceito essencial é a portabilidade do código-fonte. No entanto, conseguir código verdadeiramente portável é uma tarefa complicada e, na década de 1990, surgiram plataformas de desenvolvimento que, entre outras coisas, eliminaram a necessidade do código portável: a aplicação resultante, após a compilação, poderia ser executada em qualquer sistema necessitando, para isso, uma camada no sistema operativo que transformasse o bytecode resultante em instruções nativas.

## Contudo, com a consolidação das linguagens orientadas a objeto e ferramentas de desenvolvimento e uma maior flexibilidade e abrangência de programação trazida por elas, o conceito multiplataforma vem ganhando vertentes e se fortalencendo. Os computadores já não estão mais apenas nas mesas ou sendo carregados como livros, agora os telefones celulares podem ser computadores de bolso. Com esses dispositivos moveis, o desenvolvimento multi-ambiente ganha um outro patamar. Versões de sistemas operacionais de uma mesma empresa fabricante são desenvolvidas para diferentes tipos de maquinas. Servidores, Workstations, computadores desktop, notebooks, tablets e smartphones já podem ter sistemas distintos, adequados ao seu hardware e as necessidades do usuário, e serem ou não, de uma mesma fabricante. O mesmo acontece com softwares de aplicação, comerciais, jogos, entre outros. Para esse processo, pode-se utilizar uma ferramenta que deixe que o software fique padronizado permitindo seu funcionamento em diferentes sistemas ou atraves de diferentes ferramentas. Entre as maiores barreiras encontradas no desenvolvimento multiplataforma e a questão de uma necessidade de se padronizar o software de forma que ele tenha um desempenho satisfatório nos diferentes sistemas em que ele funcionara. Dificilmente se conseguira um desempenho total que o software pode ter em cada sistema.

O problema da descrição das interfaces das aplicações consiste na dificuldade que o engenheiro de software encontra em projetar as interfaces com usuário para que estas possam se adequar a diferentes tipos de dispositivos e executar em diferentes plataformas de programação. Para solucionar esse desafio, podemos citar os seguintes trabalhos: o UIML ( User Interface Markup Language ) [14], o Icrafter [21], o UID ( User Interface Description ) [16] , o MAB ( Mobile Application Builder ) [17] e MultiMad [75]. Esses trabalhos propõem a definição da interface utilizando documentos XML (i.e., eXtended Markup Language ) [36].

Essa definição é feita através de marcações (i.e., tags ) pré- estabelecidas para a criação dos componentes de interface. Depois, geradores de código transformam a descrição inicial em código executável ou em documentos de uma linguagem de marcação (e.g., XHTML). No caso do MAB e do MultiMad, ferramentas gráficas são disponibilizadas para facilitar a criação das interfaces das aplicações. Outros trabalhos foram propostos para prover a adaptação de conteúdo baseada nas características dos dispositivos, nas preferências dos usuários e no contexto do ambiente em que o dispositivo se encontra (e.g., mudança na largura de banda). Algumas dessas soluções utilizam proxys ou gateways para realizar a tarefa de adaptação. Contudo, esses trabalhos concentram toda a decisão da adaptação em uma arquitetura geral, podendo não ser satisfatória para certos tipos de aplicações que não consistam na simples requisição de um conteúdo Web. Além disso, essas soluções não apresentam formas eficientes de captura e gerenciamento dos perfis do usuário e do dispositivo.

Atualmente estamos em um momento em que as pessoas ao redor de mundo estão mudando a forma como agem e interagem. Esta nova forma de interação, seja com outras pessoas ou através da internet somente tornou-se possível através da mobilidade!

Com isso, precisamos desenvolver soluções para esse novo público, que está interagindo e muito com essa nova plataforma, a Mobile.

Porém, Mobile não é só um ou outro dispositivo, são todos dispositivos que trazem mobilidade, flexibilidade, onipresença e são fáceis de transportar de um lugar pra outro.

Daí vem o desafio, “Como desenvolver de uma forma que se aproveite uma solução para diferentes dispositivos?”. Há pouco tempo era praticamente impossível alcançar o resultado deste desafio, visto as diferentes plataformas existentes e seus Sistemas Operacionais, Designer e principalmente linguagens e IDE próprias.

***Vantagens***

O seu software não fica limitado a uma única plataforma O potencial de uso do software é expandido

A preocupação desde o início em desenvolver o software livre de plataforma pode poupar um bocado de dinheiro em relação ao software que é portado para mais plataformas numa versão posterior e não teve tal preocupação em seu projeto

***Desvantagens***

Aumento de custo de produção Requer mais testes para asegurar que a compatibilidade seja garantida Dificuldade em se implementar certas características devido a uma plataforma não oferecer dada característica em outra plataforma