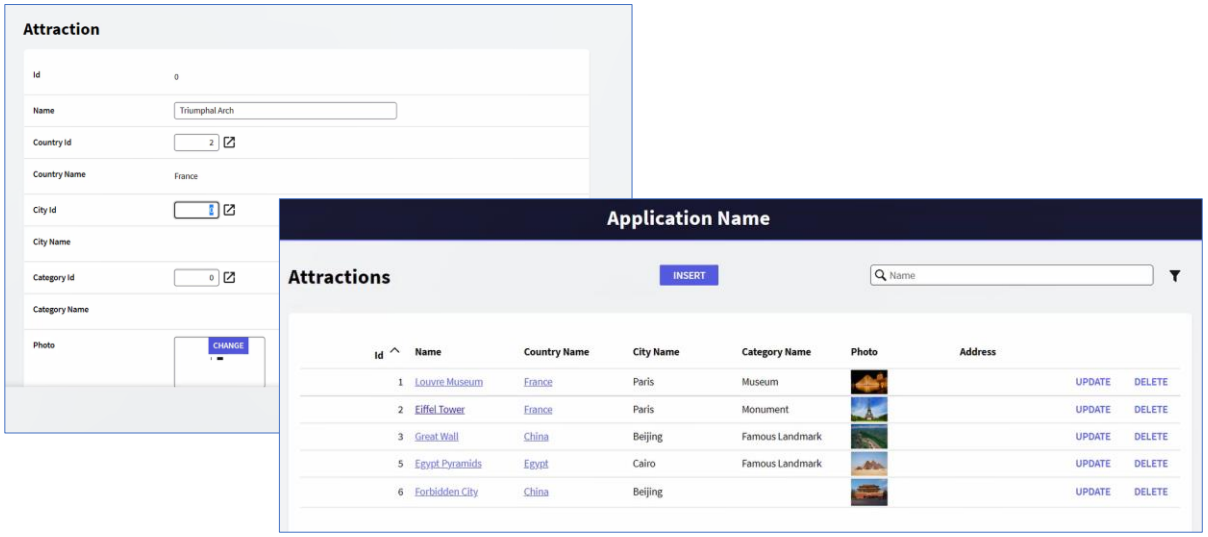
Aplicações GeneXus e sua arquitetura

GeneXus™

Aplicações web com ênfase em back-office

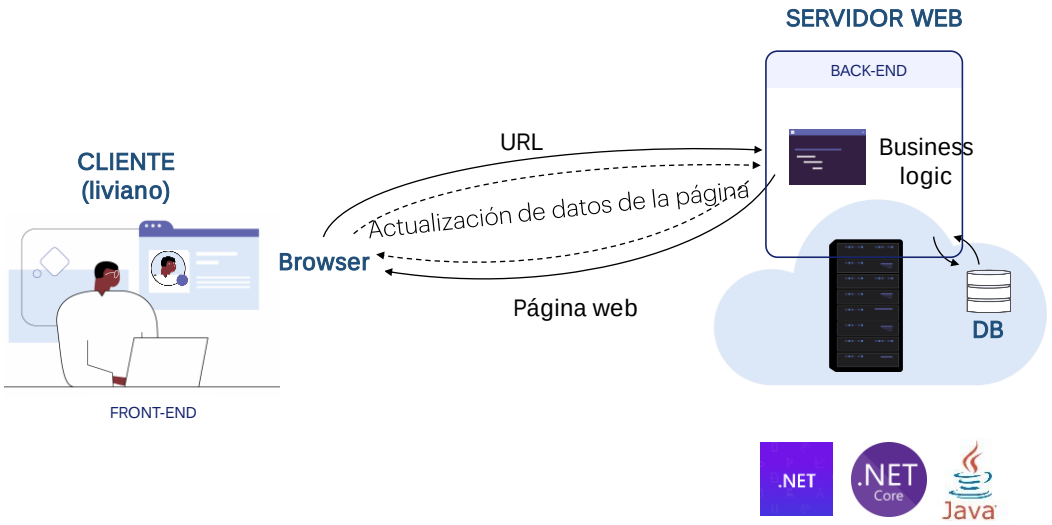


Até agora temos visto como funcionam as telas das transações, nas quais, ao sair de cada campo são validados os dados e podemos receber uma mensagem, ou que ao colocar um valor de uma chave estrangeira, se infere o nome correspondente a essa chave. Chamávamos de Client Side Validation essa resposta imediata que o usuário obtinha por parte da aplicação.

Também vimos que nas telas geradas com o padrão Work With (os objetos web panels, nos quais pararemos logo) a informação carregava-se dinamicamente quando mudávamos de página de um grid ou quando aplicávamos um filtro, ou também vimos que ajustava seu tamanho automaticamente de forma responsiva.

GeneXus

Arquitetura das aplicações web com ênfase em back-office



Algumas destas operações são realizadas na parte cliente da aplicação web (ou seja, no browser) e outra é feita no servidor, por exemplo, quando é necessário buscar informações na base de dados.

A lógica da aplicação está na parte do servidor, então quando executamos um objeto em nossa aplicação, o navegador pede a página correspondente ao servidor. O servidor prepara a informação, obtém dados da base de dados se necessário, monta a tela e a envia ao navegador para que mostre-a.

Em alguns casos, depois que a página está sendo visualizada no navegador, iniciamos uma ação como quando aplicamos um filtro a uma consulta (em um grid, por exemplo). Como consequência, o código da parte cliente se comunica com o código no servidor para que lhe devolva os dados que cumprem com o filtro, e possa depois, atualizar somente a parte da tela envolvida na ação (neste exemplo, o grid). Por sua vez, se ordenamos uma coluna de um grid ou se numa transação saímos de um campo que produz que se mostre uma mensagem, apenas o cliente a resolve, sem necessidade de acessar o servidor.

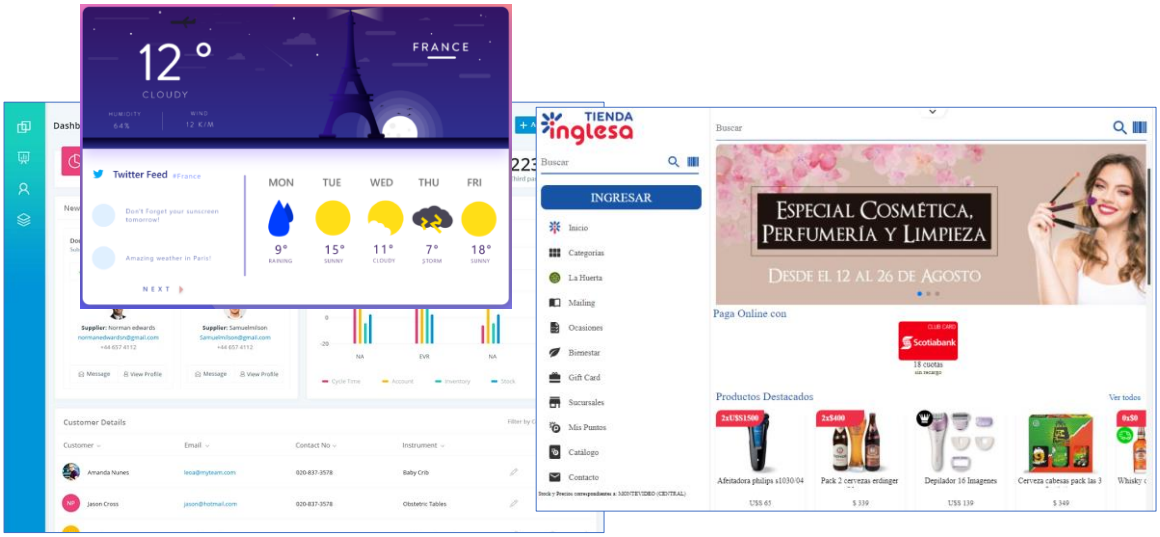
Nesta arquitetura o cliente tem alguma inteligência, mas é no servidor onde se tomam as decisões, já que ali reside a lógica completa da

aplicação.
O que desenvolvemos até o momento é basicamente o back-office da aplicação, já que vimos como manipular a base de dados (transações) e como consultar suas informações de forma mais hierarquizada, para alcançar essas entradas, saídas e modificações (ou seja, tudo o que é construído pelo pattern Work with). No entanto, com esta mesma arquitetura também podemos construir aplicações customer-facing.

Em GeneXus, para construir estas aplicações, utilizamos .Net, .Net Core ou Java, e GeneXus utiliza estas linguagens tanto para o código no cliente (front-end), como para o código no servidor (back-end).

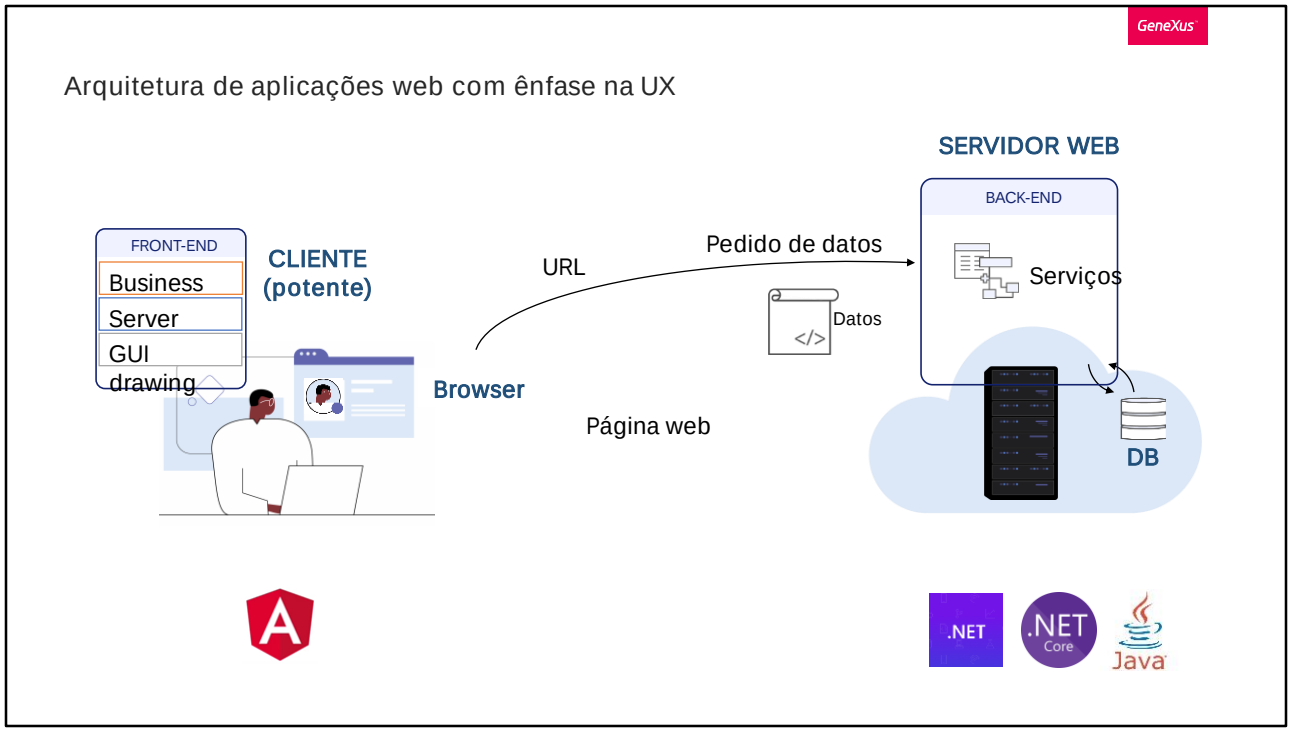
GeneXus

Aplicações de alta performance com conteúdo mais interativo



Há outro tipo de aplicações que podem ser web ou nativas para dispositivos móveis, que têm uma grande potência para mostrar informações de forma muito rápida e com foco especial no desenho e interatividade.

São aplicações do tipo customer-facing, ou seja, orientadas ao usuário, com grande riqueza de informação em tela, que vão carregando as partes da página só quando vão sendo necessárias, porque são desenhadas para uma altíssima performance com a melhor experiência de usuário possível.



Nesta arquitetura, também executa uma parte no cliente e outra no servidor, mas neste caso boa parte da lógica da aplicação está no cliente. No cliente há 3 camadas bem diferenciadas: a lógica do negócio, as comunicações com o servidor e a parte que se encarrega do desenho da tela no navegador.

Com este esquema, a maioria das funções são cumpridas no cliente e só se acessa o server para pedir dados da base de dados ou modificá-los e outros recursos oferecidos por serviços.

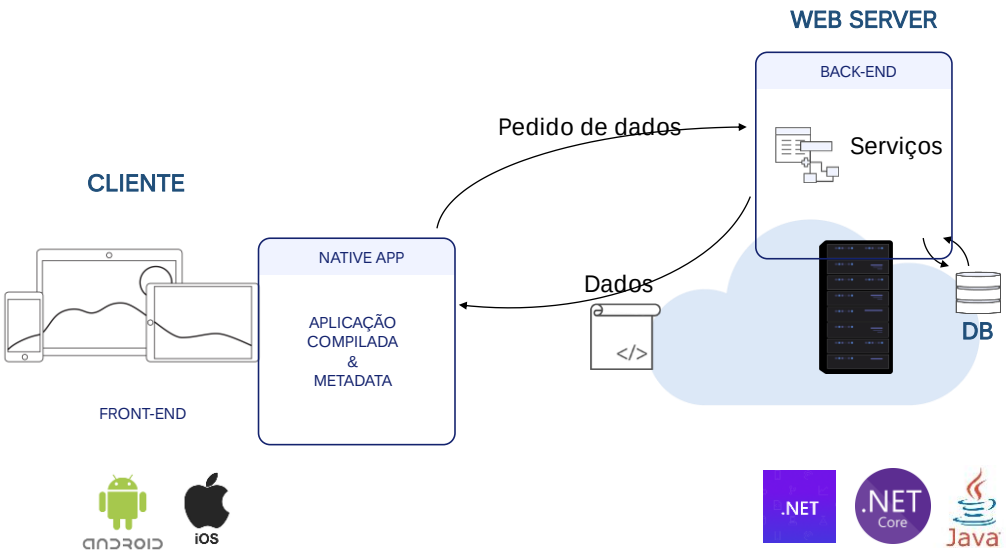
Este tipo de aplicações customer-facing que requerem a potência no cliente para poder oferecer elevados requisitos de experiência de usuário e interatividade, são mais recentes que as aplicações responsivas que vimos antes.

Para poder lidar com estas exigências, têm surgido no mercado frameworks como Angular, React ou Vue, que têm toda a potência para poder construir um cliente mais inteligente como o que necessitam estas aplicações.

No entanto, como mencionamos antes, com GeneXus também é possível desenvolver aplicações customer-facing em Java, .Net ou .Net Core,

embora a vantagem de programá-las em Angular é que os mesmos objetos painéis que desenhamos para a interface de usuário, poderemos reutilizar praticamente sem mudanças para gerar a aplicação mobile nativa, ou seja, para Android ou iOS.

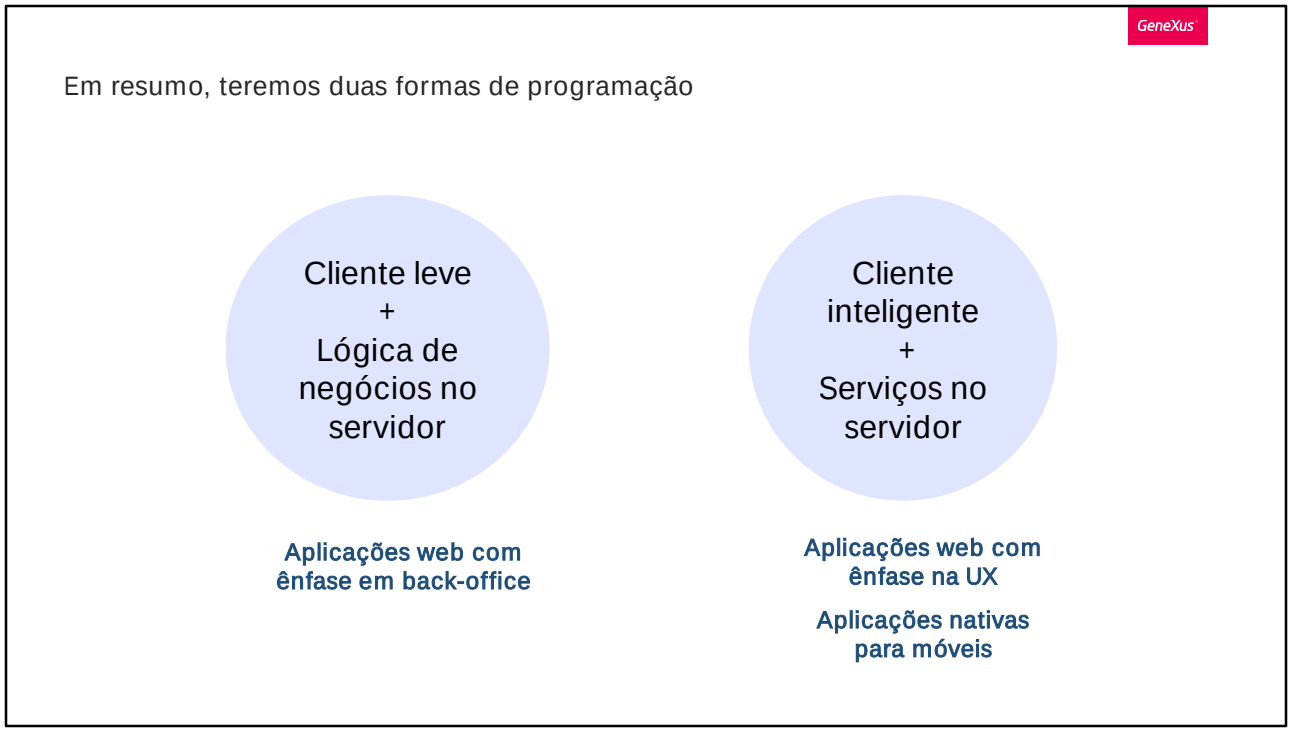
Arquitetura das aplicações móveis (online)



Até agora temos nos concentrado nas aplicações web, vejamos agora a arquitetura de uma aplicação móvel nativa.
Em particular, veremos a arquitetura de aplicações móveis sempre conectadas via Wi-Fi (aplicações on-line), que é o caso mais comum das aplicações nativas, mas sabendo que GeneXus também constrói as desconectadas.

Uma aplicação móvel também tem uma parte que executa no cliente (neste caso, o dispositivo móvel) e uma parte que executa no servidor, que fornece informações ao cliente.
O código da aplicação executa no dispositivo, que acessa o servidor apenas quando necessita dados.
Em GeneXus, podemos gerar essas aplicações para dispositivos Android ou iOS. Os serviços no server (que fornecem os dados ao cliente) podem ser gerados em Java, .Net ou .Net Core.

Como vemos, esta arquitetura é muito similar à das aplicações web de alta performance e, por essa razão, a forma de programar é muito similar, com algumas diferenças que veremos mais adiante.



A arquitetura das aplicações condicionam muitas coisas da aplicação, mas, fundamentalmente a forma de programá-las.

No caso de que o cliente seja leve, praticamente todos os requisitos da aplicação serão resolvidos no servidor, porque nossa programação sempre estará orientada a operações no servidor como, por exemplo, o acesso à base de dados.

Em vez disso, quando contamos com um cliente potente, muitas operações poderão ser realizadas no próprio cliente, sem necessidade de recorrer ao servidor. No entanto, há outras operações que têm a ver fundamentalmente com a interação com a base de dados que requer os programas do servidor.

Por esse motivo, devemos ter uma sintaxe diferente para indicar quando queremos que tal código seja executado no cliente, e quando no servidor. Isto também é válido para a programação de aplicações nativas para dispositivos móveis.

Em seguida, veremos como são programadas aplicações de cada tipo.

GeneXus™

training.genexus.com
wiki.genexus.com