

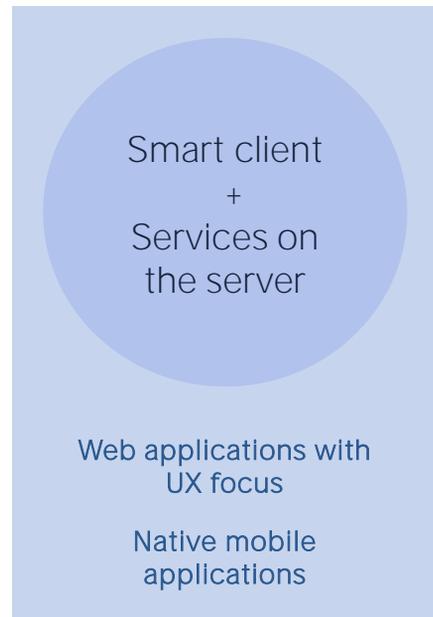
# Web Screens with Customer-facing Focus

Introduction

**GeneXus**<sup>™</sup>



Web applications with  
Back-office focus



Anteriormente mencionamos que a arquitetura das aplicações condicionava, entre outras coisas, a forma de programá-las.

Neste módulo, focaremos nas aplicações web que serão executadas em um cliente inteligente, ou seja, aplicações customer-facing que nos permitam proporcionar a melhor experiência de usuário possível.



Back-office

Web panels

### Travel Agency

Country Name: France

Attraction Name From:

Attraction Name To:

| Attraction Name                      | Attraction Photo  | Trips  |
|--------------------------------------|---|--|
| <a href="#">Louvre Museum</a>        |  | 0  <a href="#">New trip</a> |
| <a href="#">Eiffel Tower</a>         |  | 0  <a href="#">New trip</a> |
| <a href="#">Notre Dame Cathedral</a> |  | 0  <a href="#">New trip</a> |

Total Trips: 0

### Travel Agency

Attraction Name: Eiffel Tower

Country Id: 2

Country Name: France

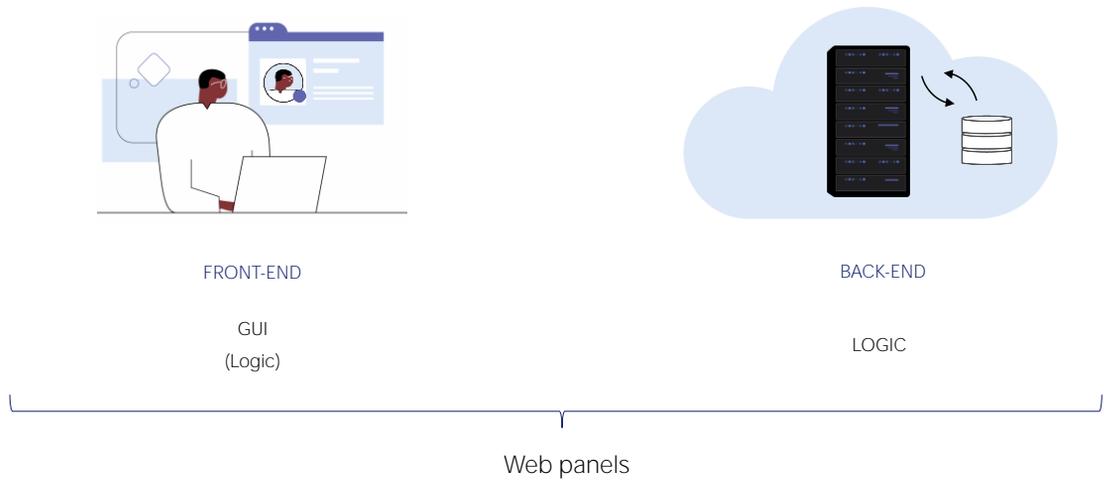
City Name: Paris

Category Name: Monument

Trips: 2

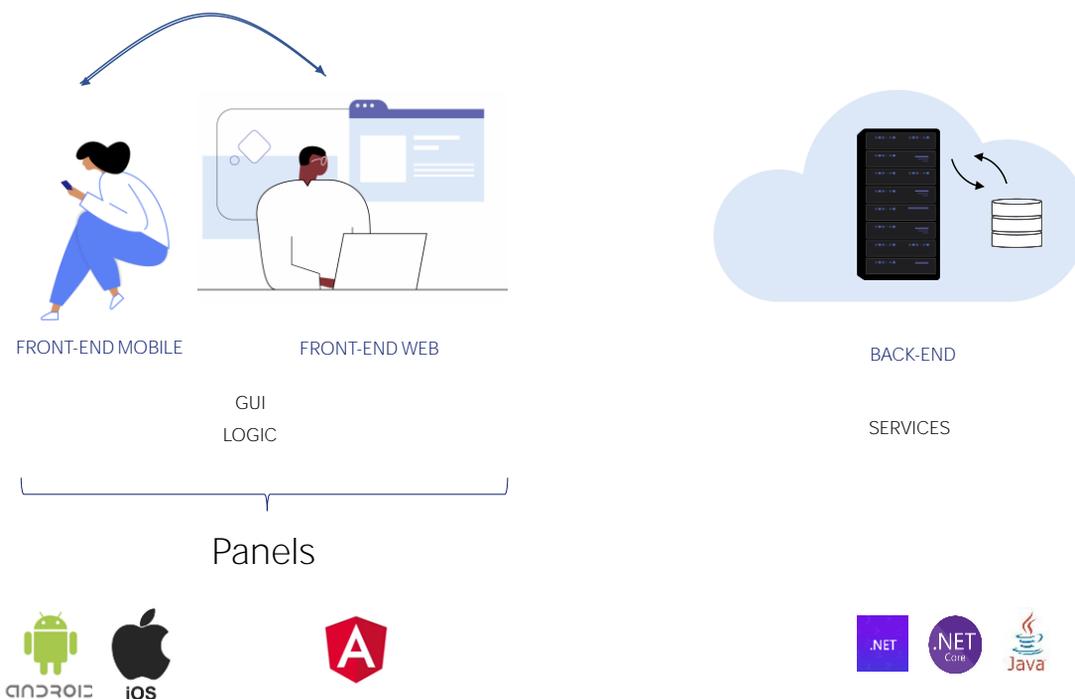


Até agora, temos visto como utilizar objetos web panels, para o desenvolvimento da aplicação web de Back-office da agência de viagens.



Também vimos como são desenhados e programados os web panels, sabendo que eles têm a maior parte de sua lógica executando no servidor.

Como agora nos interessa focar na programação com ênfase na experiência do usuário, devemos usar um objeto que nos permita que a lógica se concentre na parte cliente da aplicação.



Todas as telas com as quais o usuário interage (tanto web, como de uma aplicação nativa) tem uma parte visual que é exibida no cliente e uma lógica associada, que no caso das aplicações que nos interessam agora, as Customer-facing com a maior experiência de usuário possível, esta lógica é executada no cliente, e no servidor existem serviços que fornecem ao cliente funções relacionadas à obtenção e manutenção de dados.

O objeto GeneXus que nos permite adicionar código que se execute no cliente, assim como também código que executará no servidor através da invocação aos serviços do back-end, é o objeto Panel. Este objeto nos permitirá desenvolver aplicações customer-facing, tanto web, como aplicações móveis nativas.

Para poder dispor destas aplicações complexas do lado do cliente (front-end), utilizaremos Angular para gerar o cliente web, e Android ou iOS para um cliente de uma aplicação nativa. GeneXus nos gera a parte do servidor (back-end) em .Net, .Net Core ou Java.

Abordaremos o objeto Panel começando com sua lógica e, posteriormente, nos introduziremos no desenho da tela. O interessante deste objeto, é que as telas que desenhamos para a aplicação web poderão depois serem usadas na aplicação nativa e vice-versa, já que a forma de programar é válida para ambas as

plataformas.

Aqui, veremos somente uma introdução ao objeto Panel, em particular para o desenvolvimento de telas web. Em outros cursos, como o de desenvolvimento de aplicações web com Angular, ou o de desenvolvimento de aplicações móveis nativas, poderá aprofundar no uso deste objeto

*GeneXus*<sup>TM</sup>

[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)