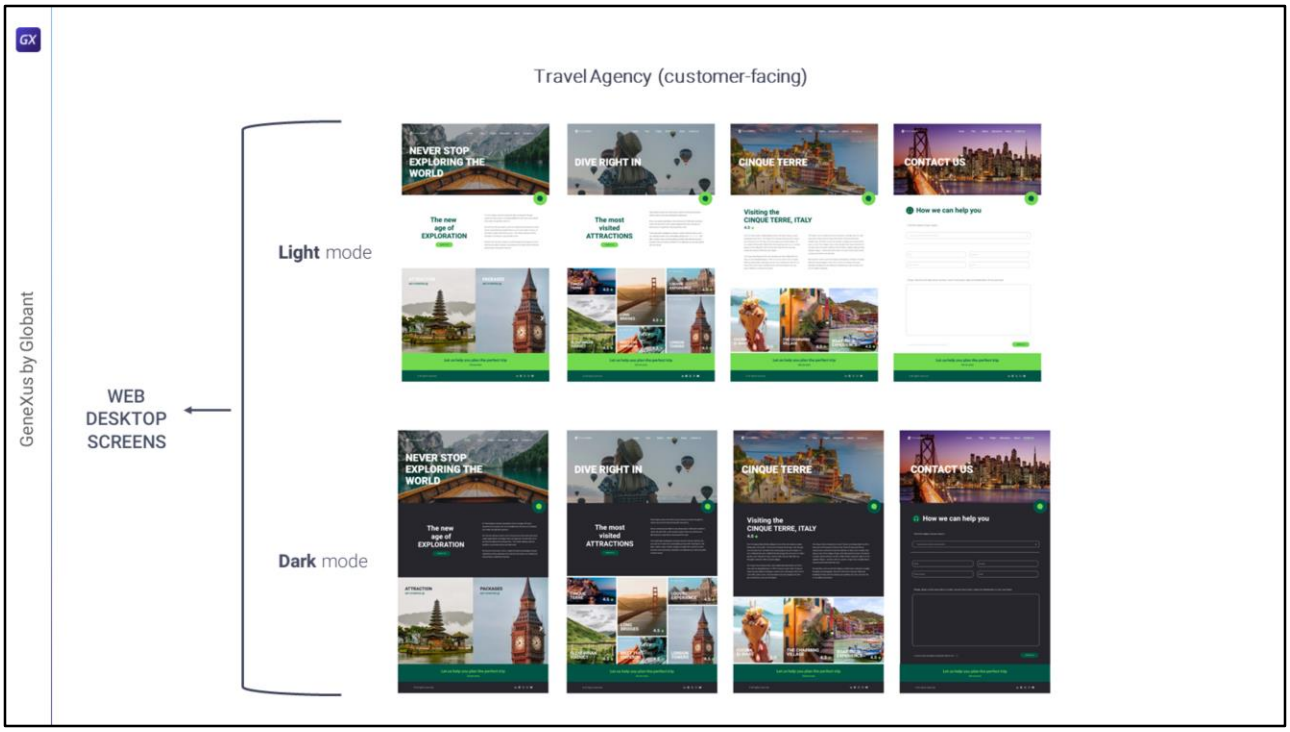
GeneXus by Globant

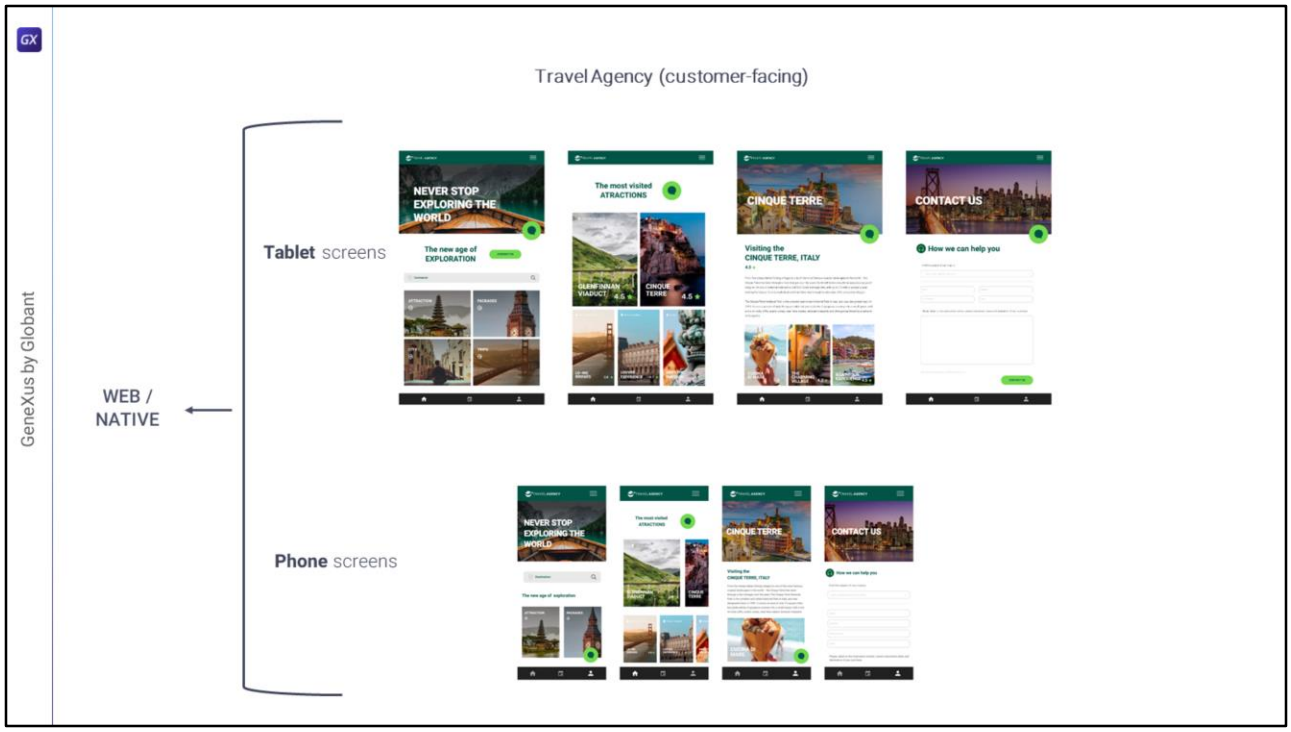
Multiexperience Approach

Cecilia Fernández

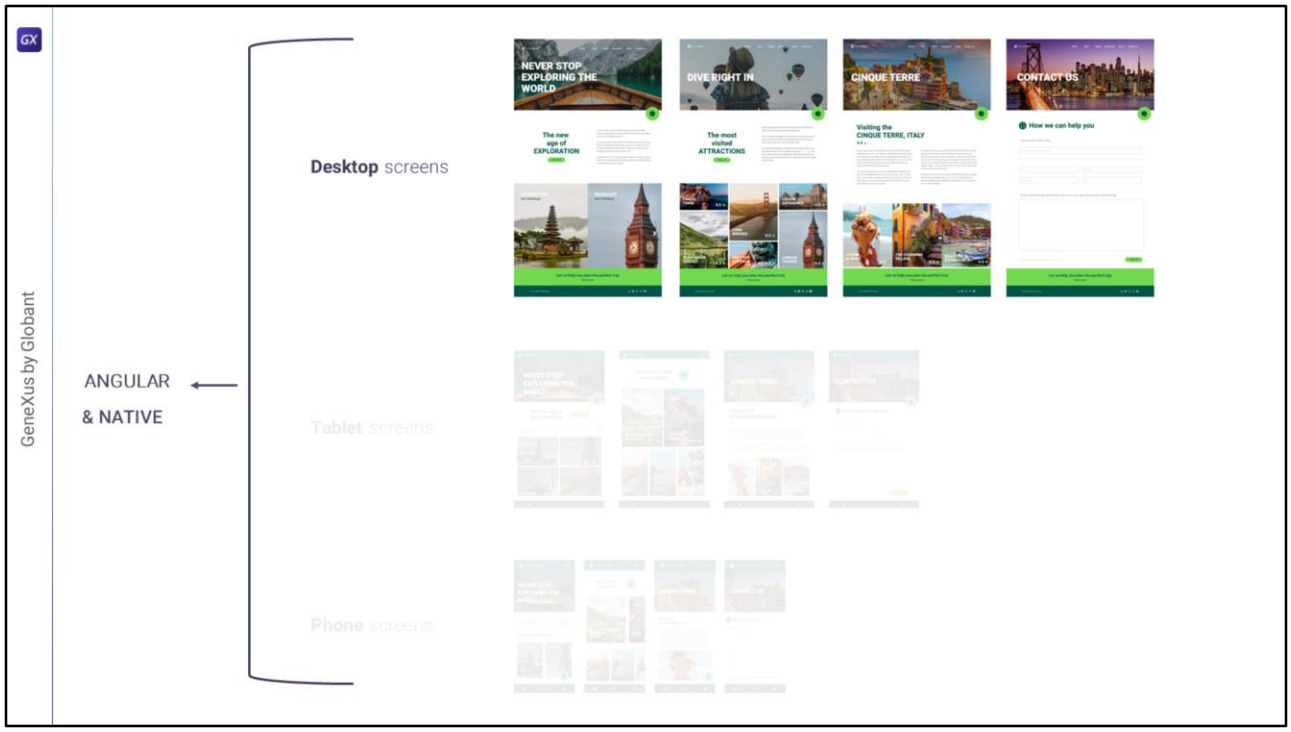


A lo largo de los cuatro módulos anteriores nos dedicamos fundamentalmente a adquirir las competencias necesarias para poder desarrollar la aplicación Customer-facing para Angular.

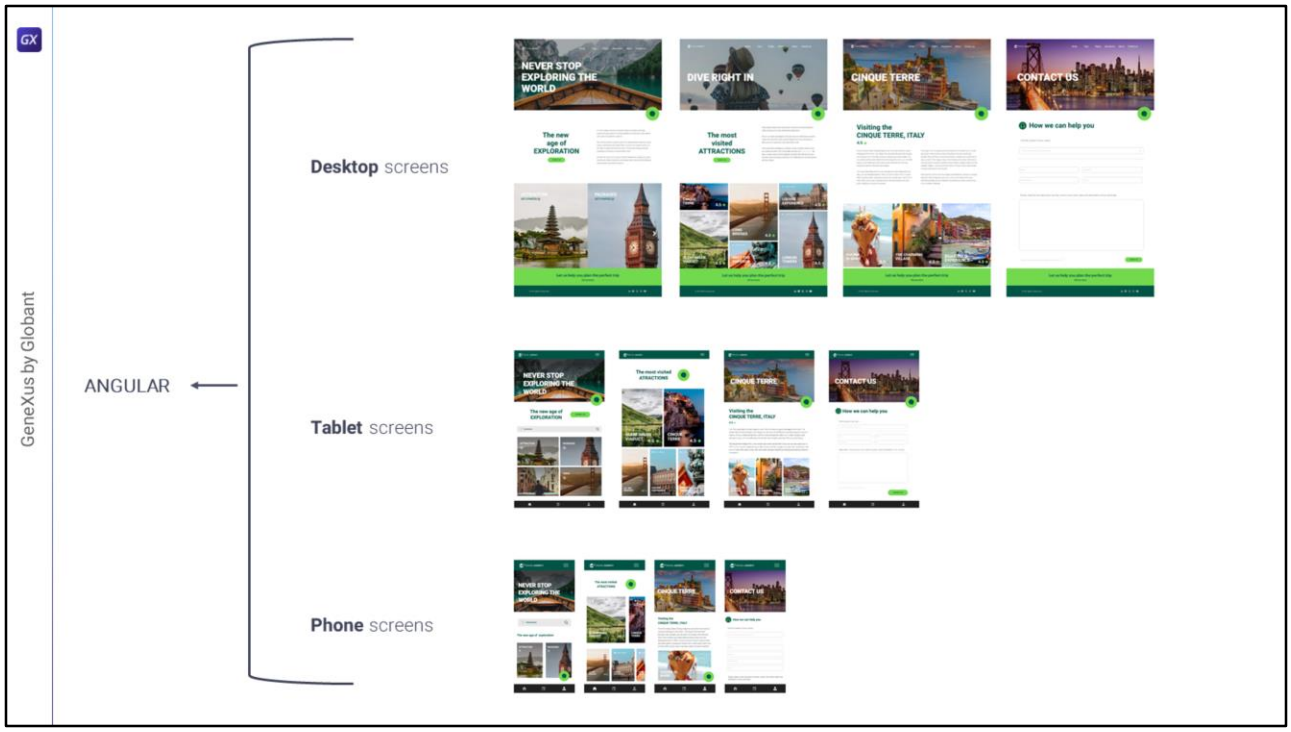
Nos concentramos casi únicamente en el desarrollo de las pantallas en tamaño Desktop…



… Pero vimos que nuestra diseñadora también nos había diseñado las variantes para tamaños Tablet y Phone.

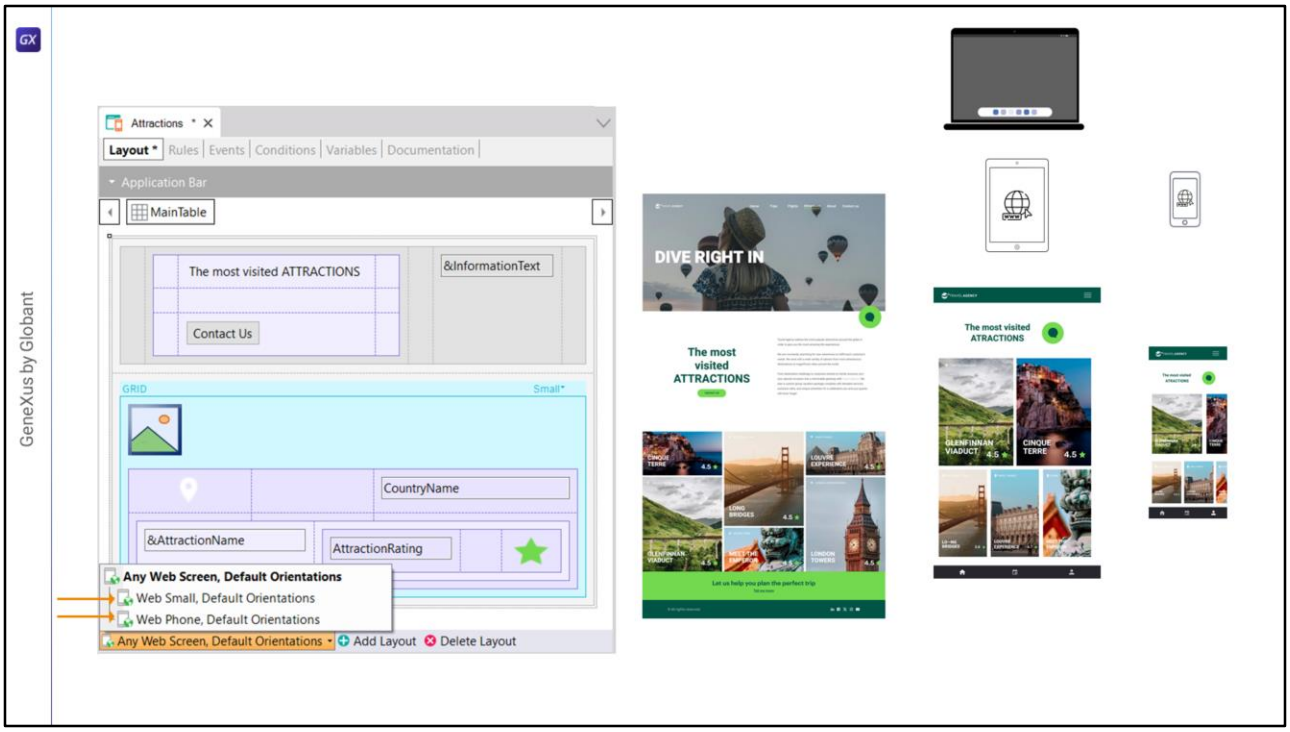


Y aquí es donde se nos complejiza el asunto, porque mientras para tamaño Desktop sólo tendremos la aplicación Angular, para tamaños Tablet y Phone sí tendremos la famosa multiexperiencia: es decir, vamos a necesitar implementar tanto la aplicación Angular como la nativa para Android y para Apple.



Pensemos por unos instantes sólo en la aplicación Angular. La misma aplicación deberá funcionar adaptándose a los distintos breakpoints, de acuerdo al diseño adaptativo que rige a esta plataforma.

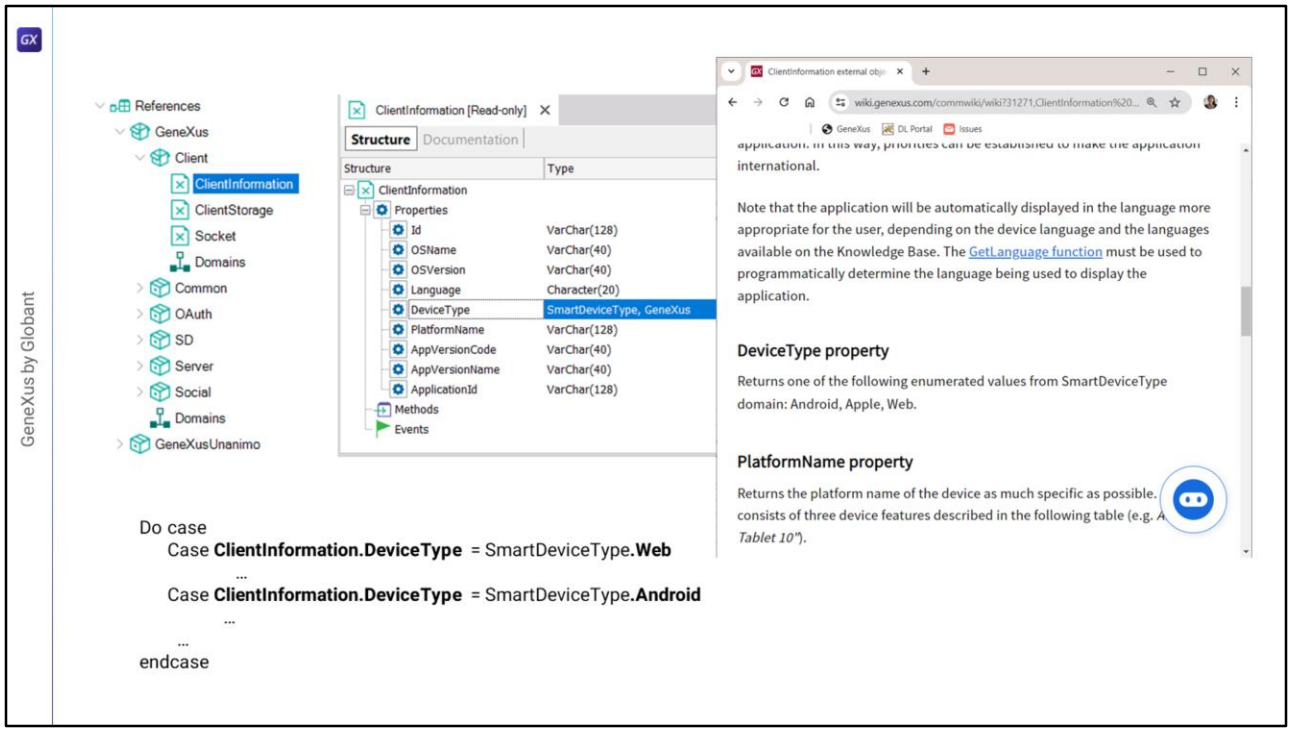
Esto se realiza automáticamente si nosotros para cada objeto definimos estos dos tipos de layouts, además del que ya teníamos.
Es decir, en total tendremos, para este caso, 3 layouts por objeto.



Así, para este panel tendremos el layout default (al que podríamos también especializar para indicar que es Any Web Screen, lo que dejaría por fuera todo lo que no sea Web), decía entonces, que tendremos este layout default que corresponderá a este diseño… y un layout para Tablet y otro para Phone.

El layout Web Small será el elegido tanto si se está ejecutando la aplicación en un dispositivo móvil que tenga el tamaño correspondiente a esa plataforma, o en una laptop de ese tamaño de pantalla. En cambio el layout Web Phone siempre corresponderá a dispositivos móviles del tamaño indicado por su plataforma.

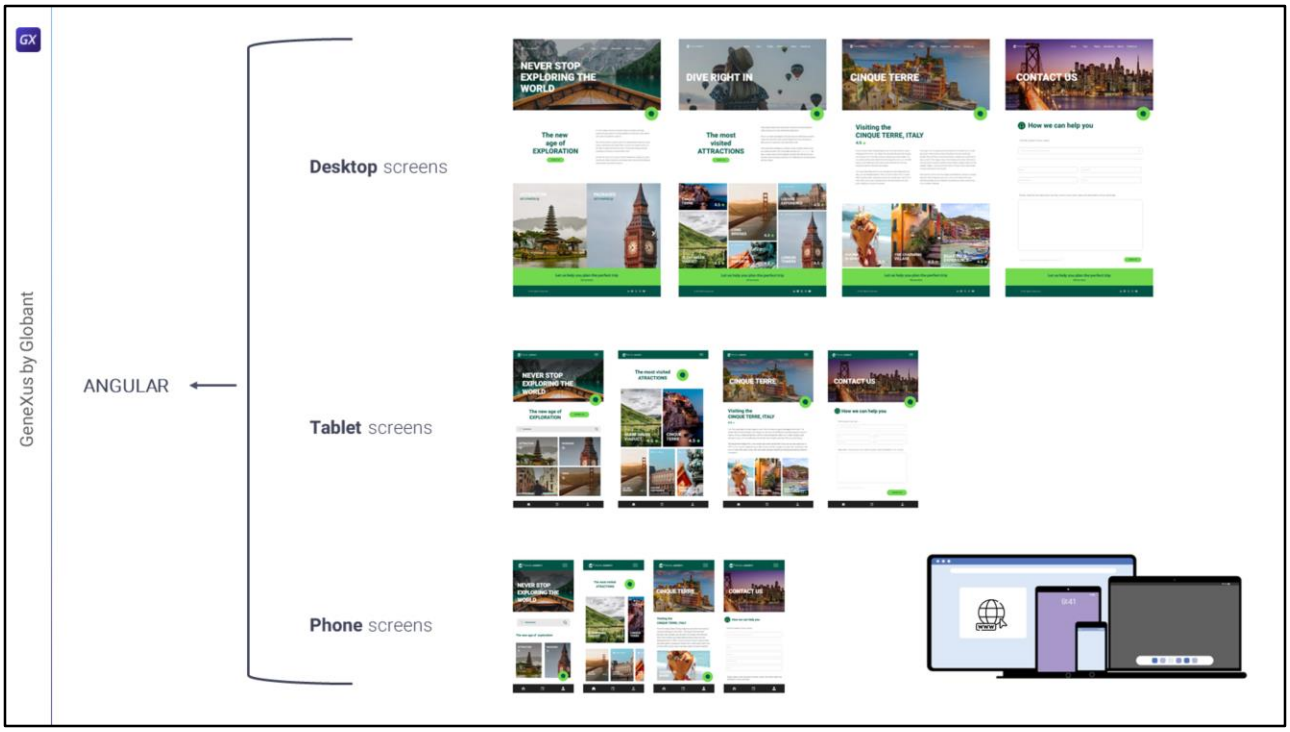
También es importante mencionar que el objeto panel Attractions será el mismo, independientemente de la cantidad de layouts que le definamos. En un único objeto implementamos todas las variaciones. Aunque solo para la sección Layout podremos establecer estas versiones distintas. Las demás secciones serán únicas para todas las variedades. Es decir, no habrá allí versiones. Los eventos que se especifiquen aquí aplicarán a todos los layouts…



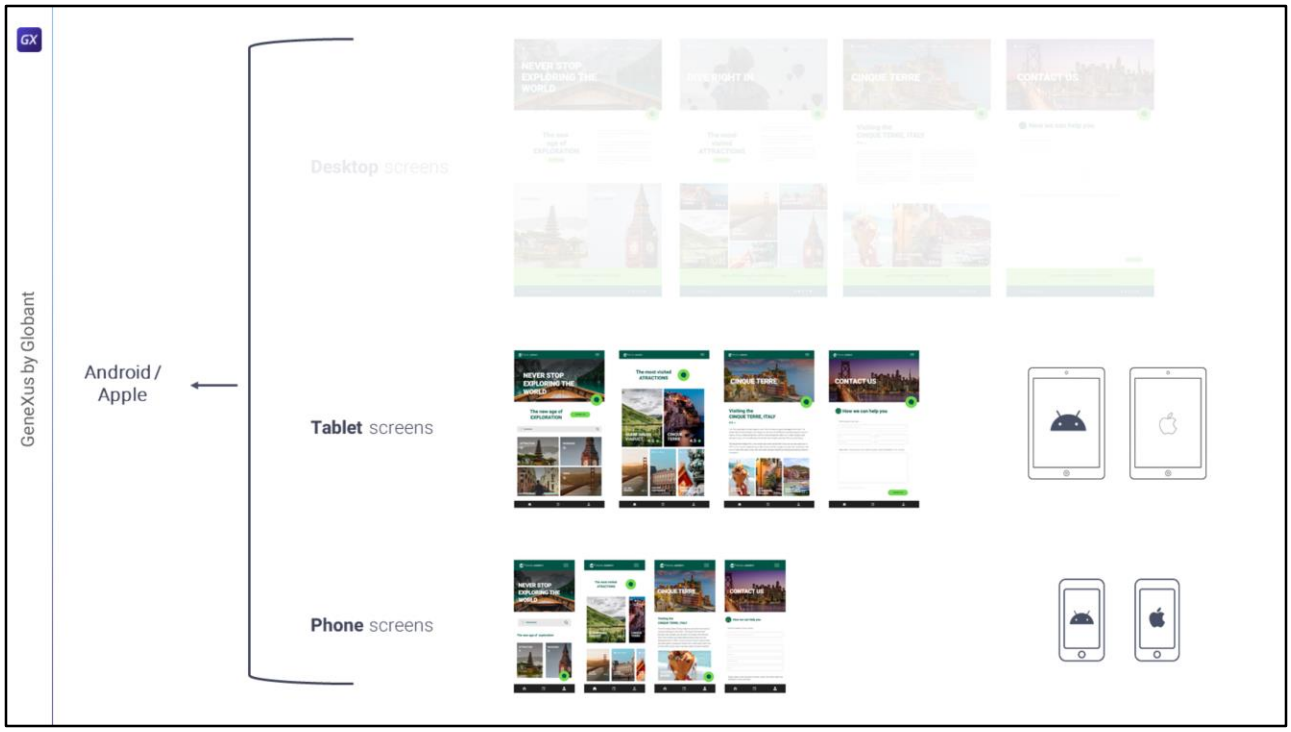
Pero tenemos la posibilidad de ejecutar un código u otro dependiendo de la plataforma. Para ello contamos con este objeto externo ofrecido en toda KB dentro del módulo GeneXus, submódulo Client.
Vean todas las propiedades que ofrece el objeto. Y en particular vean la DeviceType, tipo de dispositivo.

Preguntando por su valor podemos saber si el tipo de dispositivo que está ejecutando es Web o Android o Apple.
De esta manera podemos programar comportamiento específico en cualquier lugar donde se admita código, como los eventos.

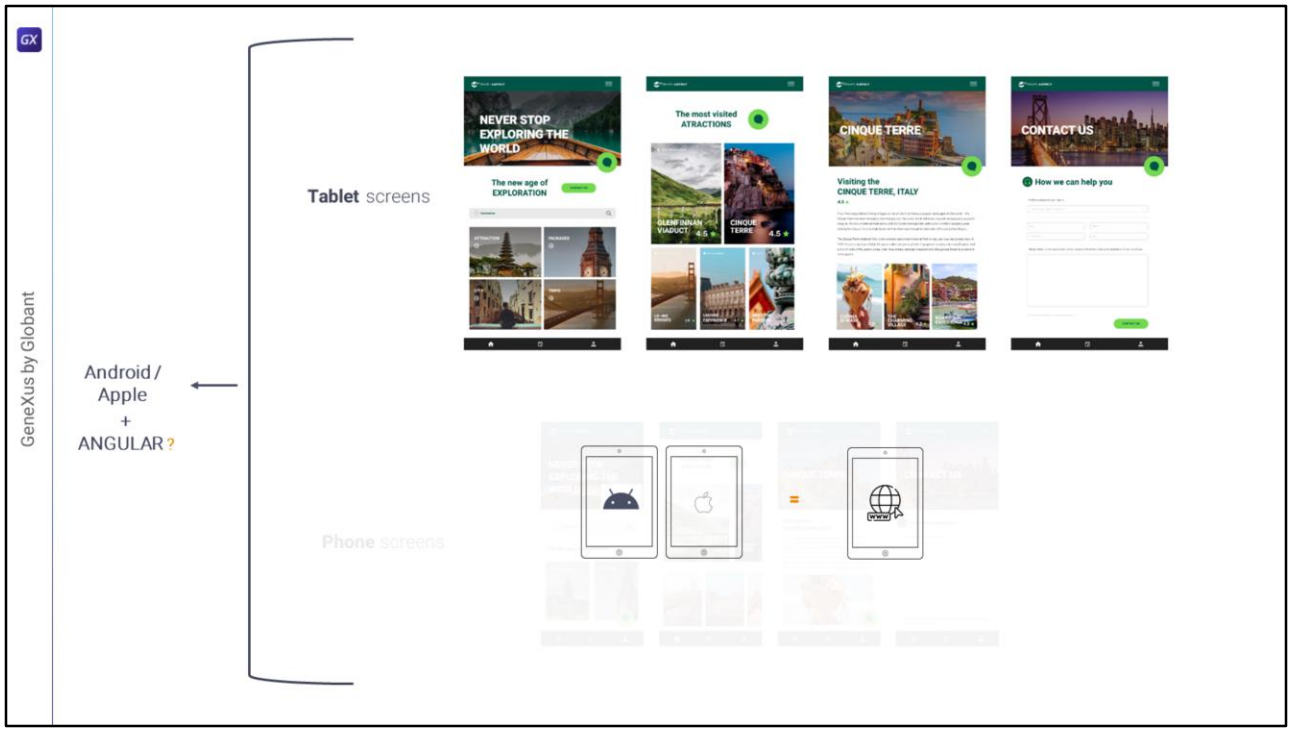
No todas estas propiedades valen para todos las plataformas, pero de esta manera podemos lograr saber algunas características del entorno en el que se está ejecutando nuestra aplicación, y tomar las acciones que deseemos.



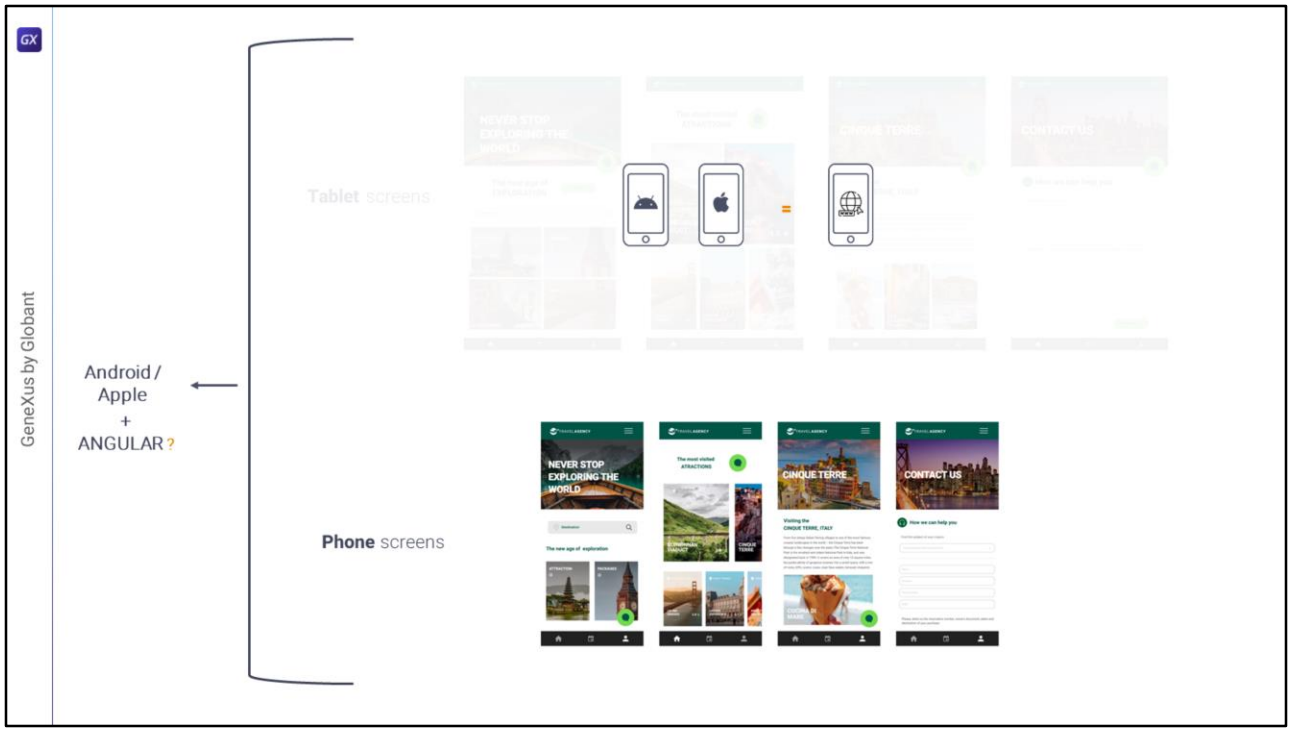
Así que tendremos los mismos 4 paneles para las tres variaciones. Y el objeto main será el panel Home. Este será el que se compilará de acuerdo a su árbol de invocaciones. Y su Url será el punto de entrada de la aplicación desde el browser de cualquier dispositivo, tanto de escritorio como móvil.



Si ahora pensamos exclusivamente en las aplicaciones nativas, usarán los mismos 4 paneles, pero…

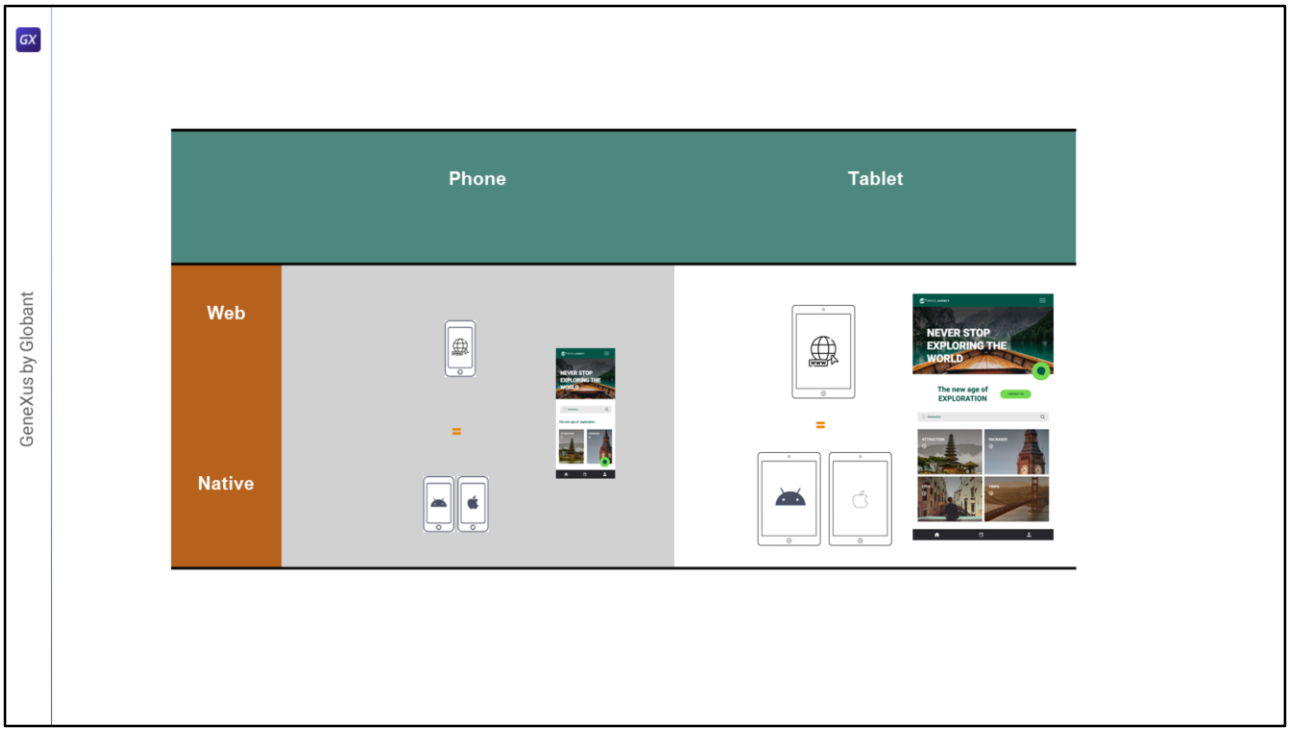


…¿las que son para Tablet, usarán exactamente los mismos layouts que su doble Web para Tablet?…



…¿Y las de Phone los mismos layouts que la de Phone Web? ¿y la implementación en GeneXus podrá ser la misma exactamente?

Sería genial, para economizar y reducir las duplicaciones y las desventajas que éstas producen. Pero, ¿se podrá?

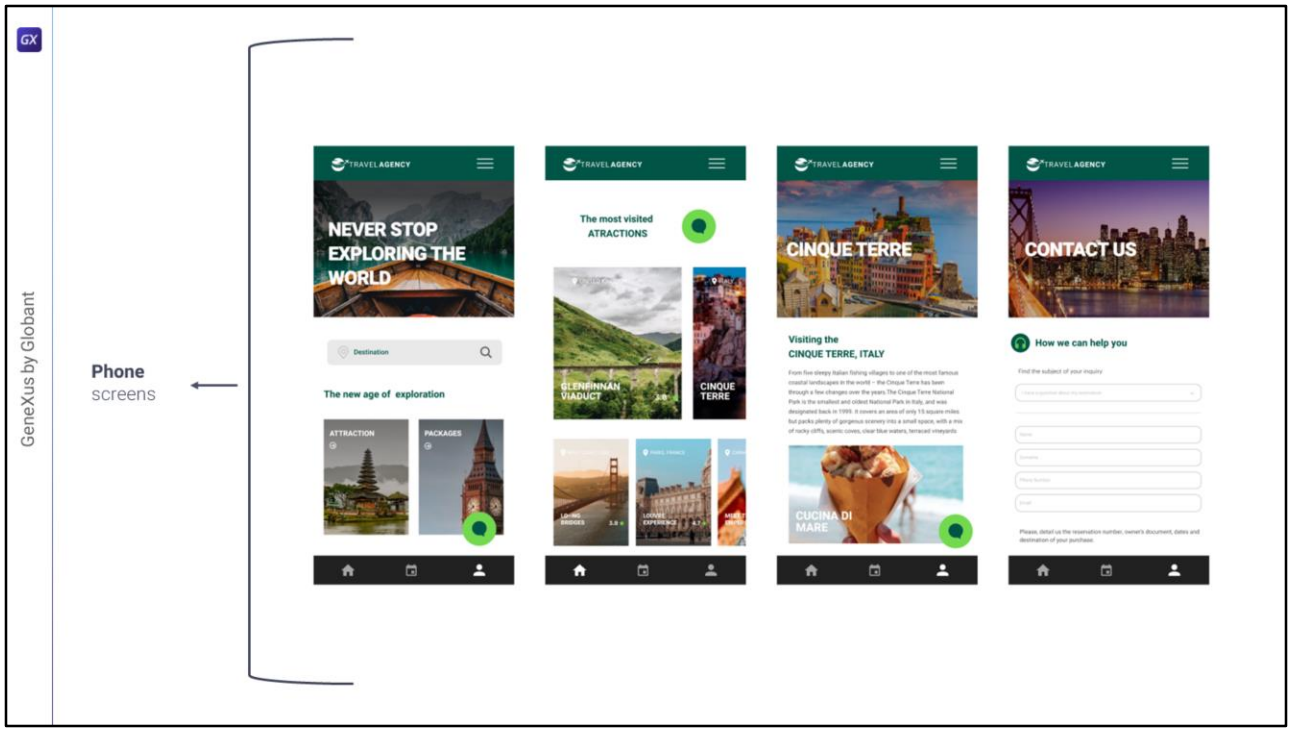


La respuesta es sí y no.

Lógicamente para reutilizar lo máximo no podremos resolver cada escenario al mejor modo de su plataforma, sino que deberemos sacrificar algunos aspectos que serían más naturales para esa plataforma en favor de una solución que sirva para cubrir también la otra. Porque la idea será cubrir lo más posible desarrollando lo menos posible.

Dicho de otro modo: o elegimos la plataforma con la que vamos a empezar, desarrollamos en esa plataforma de acuerdo a sus prácticas más naturales y recién después pensamos la otra plataforma, o ya desde el vamos pensamos una solución que se adapte lo mejor posible a ambas. Esto, desde ya, también vale para el diseño.

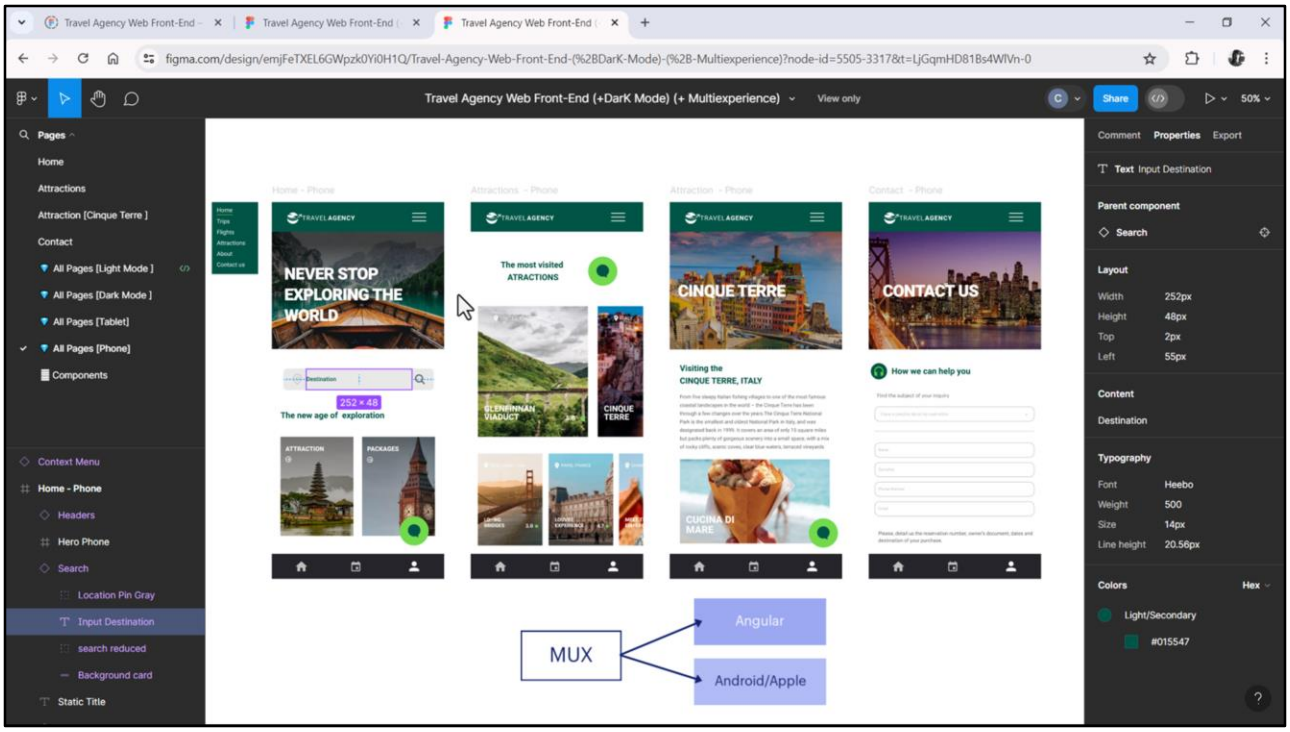
En definitiva, la gracia sería poder tener el mismo layout para tamaño Phone para web y para la aplicación nativa, y la programación lo más parecida posible también. Y lo mismo en lo que hace al tamaño Tablet.



Pensémoslo para el caso Phone, ya que el Tablet será análogo.

La pregunta es si nos conviene buscar la solución más transversal, si se nos ocurre una que valga para Web y para Android y Apple.

Para ello deberemos tener más o menos claras las diferencias entre la implementación web y la nativa.



Así, por ejemplo, implementar todas estas partes de las páginas parecería en principio indistinto. Parecería a priori que si implementásemos estas partes para el layout para tamaño Phone para Angular, los controles y sus clases y los DSOs, todo serviría exactamente igual por ejemplo para la aplicación nativa para Android.

Sería lo deseable, pero por ahora, si bien es un horizonte para el equipo de desarrollo de GeneXus, aún no se ha concretado del todo esa integración, esa transversalidad deseada, digamos. Lo que quiero decir, y ahora se los voy a ejemplificar, habrá también algunas diferencias, fundamentalmente en lo que hace a las propiedades de las clases, que, mientras para el mundo Web incluyen a todas las CSS, para el mundo nativo no se cuenta ni con HTML ni con CSS, por lo que algunas valdrán exactamente iguales, otras tendrán otros nombres u otros valores, y otras directamente no existirán. Eso, claro, nos trae aparejado un problema.

Como les decía, para lograr decidir qué nos conviene más para un desarrollo multiexperiencia, necesitamos conocer las particularidades de cada uno de los mundos.

Nosotros empezamos el desarrollo en los primeros módulos de este curso metiéndonos en el mundo Angular, sin atender para nada el mundo nativo. Y eso nos llevó a tomar algunas decisiones que ahora nos cuestionaremos, con algunos ejemplos concretos.

Seguimos en el próximo video.


GeneXus by Globant
GeneXus™
by Globant
training.genexus.com