

## Desarrollo de Aplicaciones Web-Parte 2

### Aplicaciones Web Mobile

Veamos ahora las aplicaciones Web Mobile, es decir, las aplicaciones Web que se ejecutarán en los navegadores de los Smart Devices.

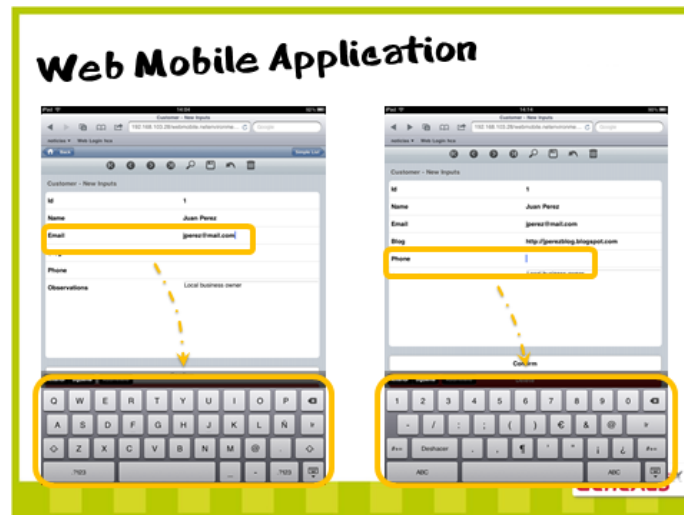
Desde el punto de vista del desarrollador se puede pensar que consiste simplemente en hacer un conjunto de páginas web que se ejecuten en esos dispositivos, páginas más livianas, más sencillas, pero en definitiva es la misma aplicación.

Eso puede ser una primera y rápida aproximación al tema, de hecho ya en GeneXus eso lo podemos hacer, por lo tanto, se puede pensar, qué nos ofrece esta nueva versión de GeneXus para aprovechar la ejecución de la aplicación en un Smart Device?



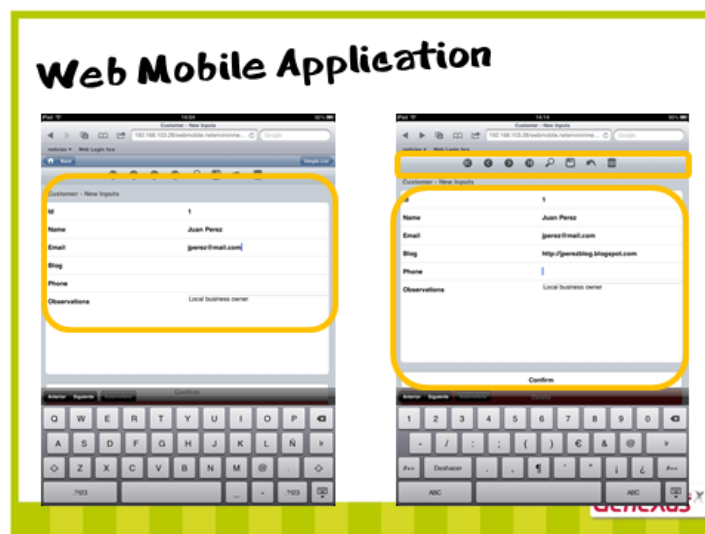
Cuando accedemos a una Aplicación Web desde un Smart Device, se pueden apreciar detalles de la integración nativa del dispositivo con HTML 5, por ejemplo, cuando se está editando una página, y se está ingresando una dirección de correo electrónico, se muestra un teclado optimizado para el ingreso de esta dirección.

De la misma forma, para el ingreso de un campo numérico, como un teléfono como es en este caso, aparece un teclado numérico.



Además, se tienen que tener en cuenta ciertos aspectos para el diseño de este tipo de aplicaciones, como por ejemplo:

- Las fuentes, con el fin de emular más aún el uso de una aplicación nativa para Smart Devices
- La utilización de un Theme especial para diseñar las páginas, que deben ser más sencillas y de tamaño más pequeño
- Y notar también los colores, el uso de tonalidades específicas para este tipo de aplicaciones.



La aplicación debe estar optimizada con el espacio correcto entre los campos para poder hacer tap correctamente y así acceder al uso Dominios Semánticos.

Por ejemplo, en este caso que al dar tap sobre el teléfono, nos lleva automáticamente a realizar la llamada a dicho número.



Con la versión X Evolution 2 de GeneXus podemos entonces lograr todos estos puntos que mencionamos, gracias a la integración de HTML5 para la generación de las aplicaciones Web.

Veamos ahora la Integración entre Diseñadores y Desarrolladores GeneXus



También en esta versión, se mejora la integración entre los diseñadores y los desarrolladores GeneXus, incluyendo nuevas funcionalidades asociadas al Theme, para poder generar un CSS a partir del Theme y enviarlo a los diseñadores para modificarlo.



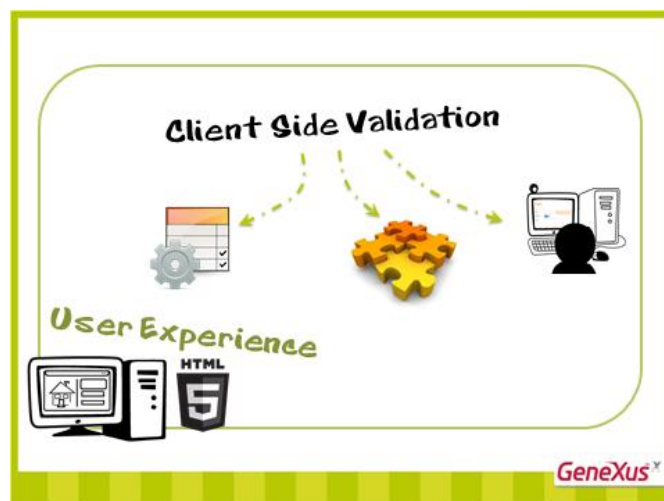
El último punto que veremos, relacionado con la Experiencia del Usuario, tiene que ver con los cambios realizados en la Validación a Nivel del Cliente para aplicaciones Web.

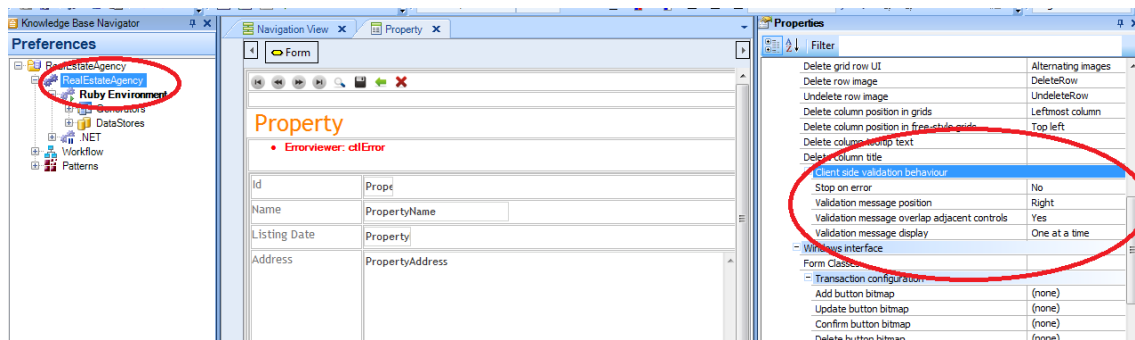
Esta funcionalidad se ha potenciado, de forma que ahora es:

- Más configurable
- Es Extensible
- Y tiene una Interacción mejorada con el usuario final.

Para poder controlar la forma en que GeneXus despliega los mensajes en pantalla se creó una nueva categoría de propiedades "Client Side Validation Behaviour" a nivel de Versión.

Veamos entonces como funcionan estas propiedades.





Estamos viendo entonces las propiedades a nivel de la versión, en particular, el grupo Client Side Validation Behaviour.

Aquí podemos ver el conjunto de propiedades que se agregaron para configurar el comportamiento de la validación a nivel del cliente.

- **La propiedad Stop on error:** para cada uno de los mensajes que se despliegan, esta propiedad indica si se puede seguir ingresando datos por más que se detecte el error, o si se tiene que reevaluar el dato. Por defecto, está en No, para que el usuario final pueda continuar ingresando datos.
- **La propiedad Validation message position:** es la posición del mensaje, usaremos el valor por defecto, que quede a la derecha del campo
- **Luego, la propiedad Validation message overlap adjacent controls:** permite indicar si el mensaje se superpondrá con otros campos o no. Al usar el valor No, se desplazarán todos los campos, para hacer lugar al mensaje.
- **Y luego, Validation message display:** Indica si se muestran todos los mensajes de error simultánea o individualmente, al momento de confirmar la Transacción.

Ejecutemos entonces nuestra aplicación y veamos como se comporta de acuerdo a estas propiedades que configuramos, al ejecutar las reglas de la Transacción Property para chequear que no quede en blanco el nombre de la Propiedad y la Dirección.





```
Navigation View x Property x
1 /* Generated by Work With Pattern [Start] - Do not change */
2 [web]
3 {
4   Parm(in:&Mode, in:&PropertyId);
5
6   PropertyId = &PropertyId if not &PropertyId.IsEmpty();
7   NoAccept(PropertyId);
8   NoPrompt(PropertyId);
9
10  NeighborhoodId = &Insert_NeighborhoodId if &Mode = TrnMode.Insert and not &Insert_N
11  NoAccept(NeighborhoodId) if &Mode = TrnMode.Insert and not &Insert_NeighborhoodId.1
12  /* Generated by Work With Pattern [End] - Do not change */
13  Default( PropertyListingDate, &Today);
14
15
16  Error( 'Please type the Property Name')
17  L   if PropertyName.IsEmpty();
18  Error( 'Address must not be empty')
19  L   if PropertyAddress.IsEmpty();
20
21  Equal(UserId, GetUser());
22  |
```

Desde el Work With Property, vamos a la opción de insert para dar de alta una nueva Propiedad.

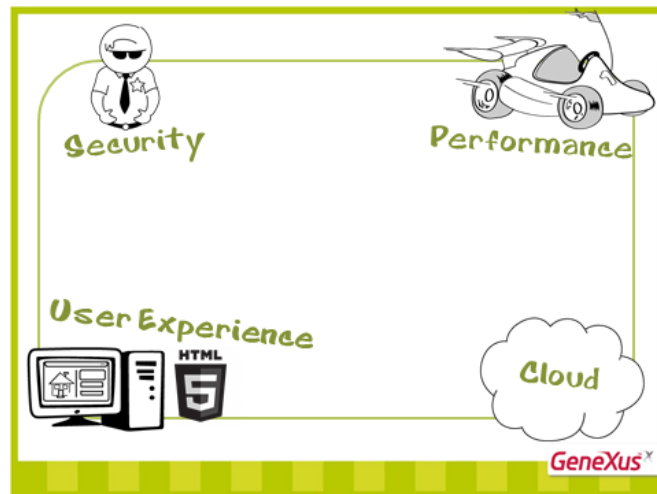
Observen que si dejamos el nombre en blanco aparece el mensaje de error, pero no nos impide que continuemos ingresando la información.

El mensaje de error no se posiciona sobre ningún otro campo, y al momento de confirmar, aparece para poder completar la información que falta.

## Property

Id	0
Name	<input type="text"/> Please type the Property Name
Listing Date	<input type="text" value="03/14/12"/> 28
Address	<div></div>
Geolocation	<input type="text"/> 
Front Image	<div></div>
Operation	Sale 
Neighborhood Id	0 
Neighborhood Name	

Hasta aquí hemos visto las grandes áreas sobre las que se ha trabajado en la nueva versión de GeneXus.



Y como se aprovechan las ventajas que brindan los nuevos estándares HTML5 y CSS3.



Y así continuar evolucionando y potenciar al máximo las aplicaciones Web generadas con GeneXus!

✓ **More Powerful Web Applications!**

GeneXus<sup>®</sup>