

## Modelo conceptual de Smart Devices

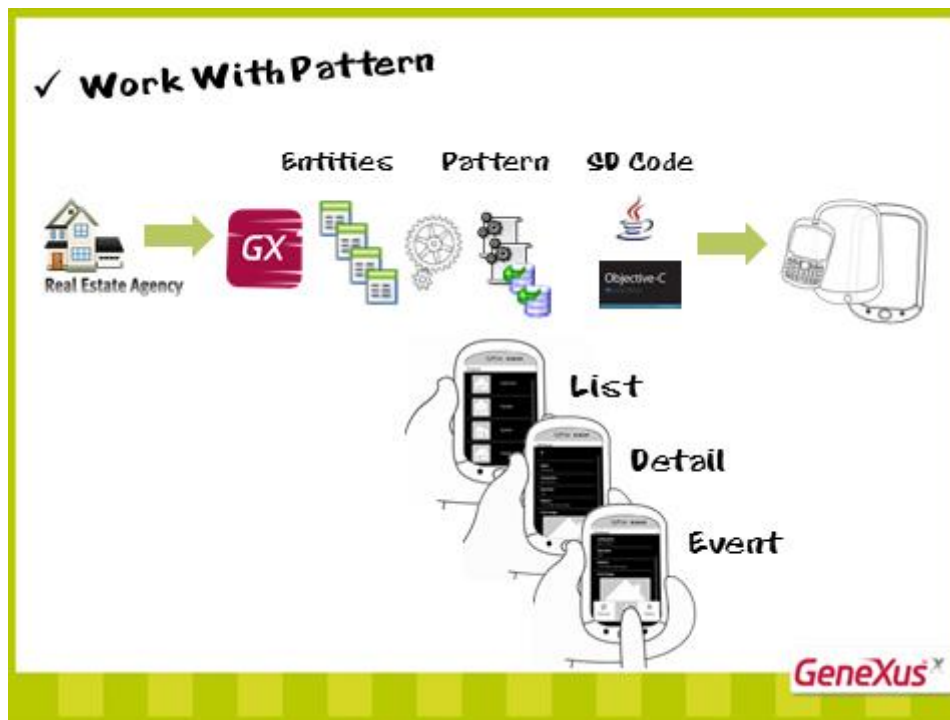


Las aplicaciones para Smart Devices organizan y manipulan la información estructurada de las **entidades** de la realidad con las que se trabaja (por ejemplo, la realidad podría ser la de una agencia inmobiliaria).

Subyace una **estructura de interacción** que podemos conceptualizar de la siguiente forma:

- Por cada entidad (por ejemplo “**propiedad inmobiliaria**” representada por la **transacción** correspondiente) se puede:
  - Elegirla de un menú y visualizar la **lista** de elementos que la constituyen en ese momento (“lista de propiedades inmobiliarias” en nuestro caso)
  - o ver la información **detallada** de un elemento en particular de la lista (en nuestro caso la información de una propiedad) y realizar alguna **acción** sobre el elemento (como por ejemplo actualizarlo).

¿No era principalmente esto lo que implementábamos con el patrón Work With de los modelos GeneXus Web?



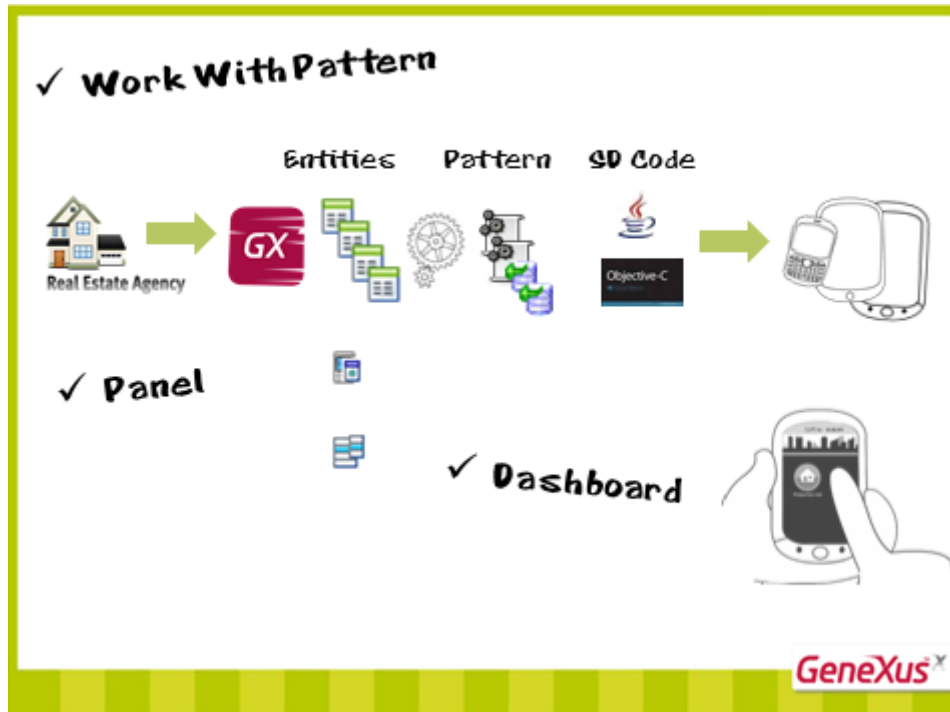
Podemos ver las **entidades** de la realidad (por ejemplo, la de la agencia inmobiliaria) como **recursos** que necesitamos manipular.

Cobrarán importancia en este modelo las **transacciones** como business components, los **data providers** que devuelven datos estructurados y los **procedimientos**.

Desarrollar una aplicación para Smart Devices implicará principalmente trabajar con el **patrón** work with específico para este generador, que a partir de las entidades (transacciones) elegidas construirá todo lo necesario para manipular la información: esto es, construirá el **listado**, el **detalle** de un elemento, y las acciones ante un evento disparado por el usuario...

en los **lenguajes** específicos de cada plataforma.

---

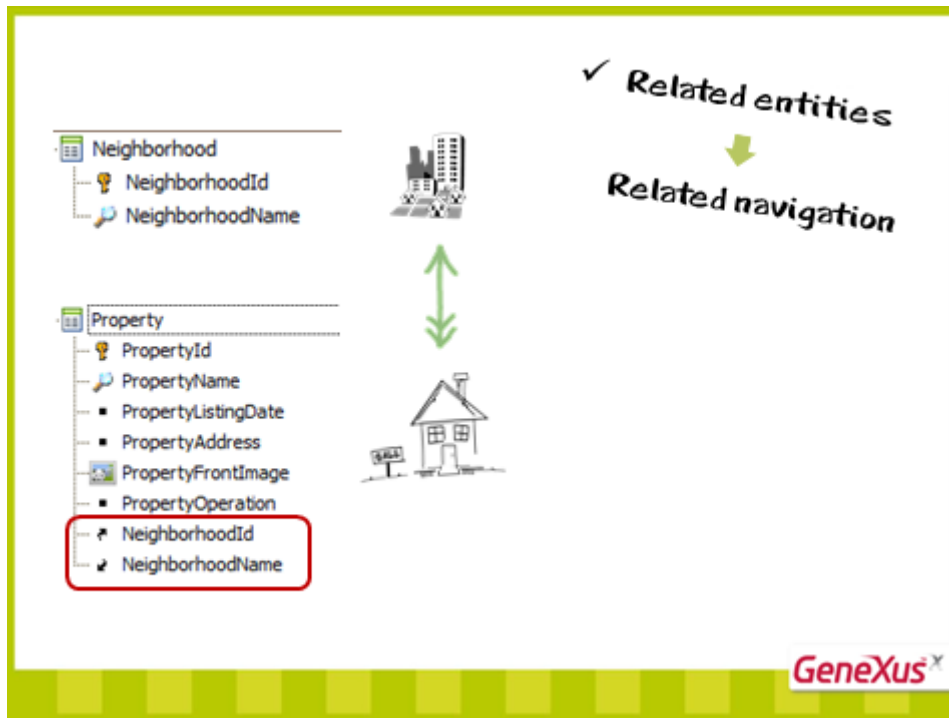


Por otra parte necesitaremos de un objeto del estilo de los generados por el work with, pero más flexible, no necesariamente asociado a una transacción, que nos permita por ejemplo pedir datos al usuario, como una fecha y realizar alguna acción. Será un objeto similar a un web panel.

Y por último, necesitaremos de los dashboards, objetos que cumplen la función de menús, utilizados principalmente como punto de entrada de la aplicación.

Resumiendo: implementar una aplicación para Smart Devices con GeneXus implicará trabajar con el patrón Work With para navegar por la información relacionada y manipularla, crear paneles para hacer consultas, desplegar información o pedir datos al usuario tal como se hace con los paneles web, y crear menús.

---



Por último, tal como ocurría en las aplicaciones web, el modelo conceptual automáticamente vincula las navegaciones de entidades que se encuentran estructuralmente interrelacionadas.

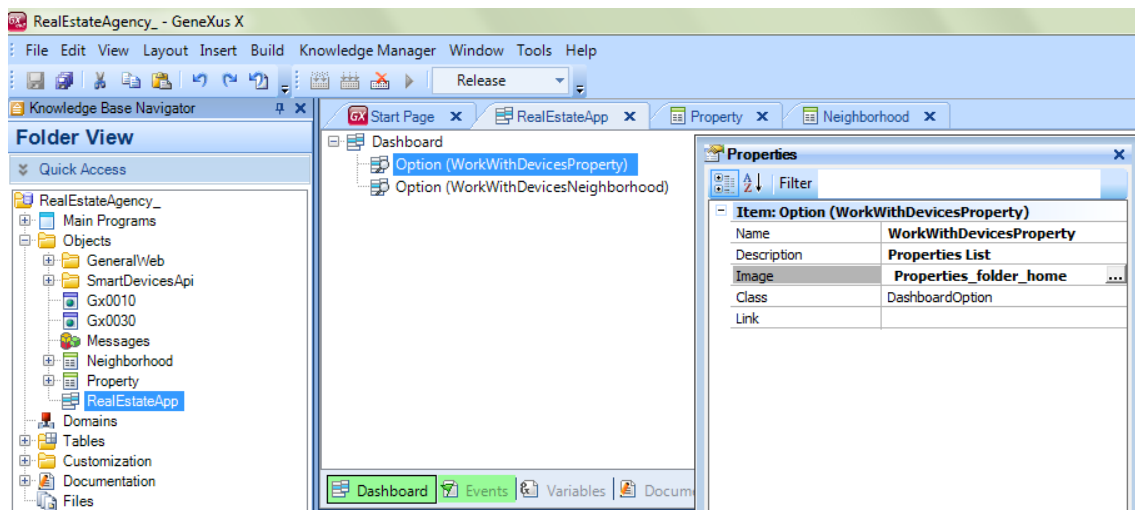
Por ejemplo, teniendo definidas las transacciones **Neighborhood** y **Property**, para representar respectivamente los vecindarios con los que trabaja la inmobiliaria y las propiedades inmobiliarias que maneja, cuando ambas entidades se relacionan, automáticamente el pattern work with lo refleja...

Veámoslo...

---

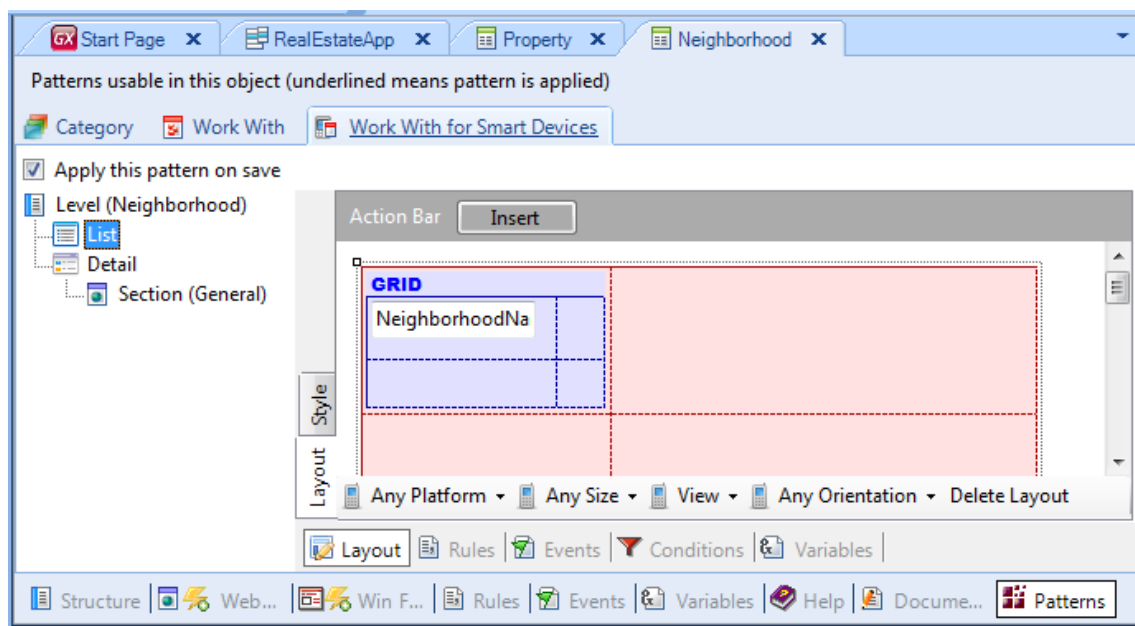
En nuestra KB tenemos las transacciones **Property** y **Neighborhood**, por el momento no relacionadas. Tenemos aplicado el pattern **Work With** en ambas.

Y tenemos un objeto **Dashboard**, que invoca por un lado al **Work With** de propiedades inmobiliarias (vemos que está invocando al Level **Property** y específicamente dentro del **Work With Level Property**, al list... y por otro lado vemos que está invocando también al **work with** de vecindarios. También a su nodo **List**.)

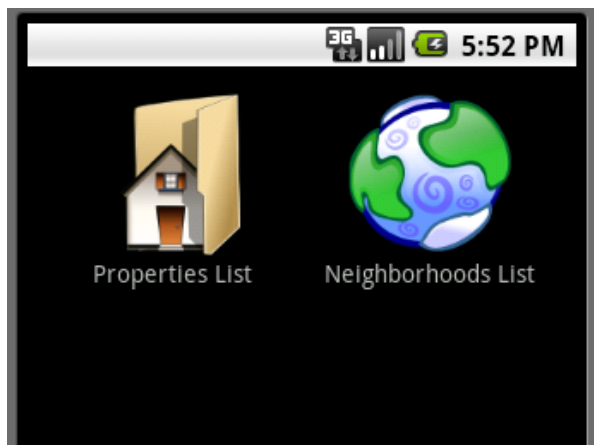


Observemos detenidamente este work with para ver cómo cambiará cuando interrelacionemos luego las entidades.

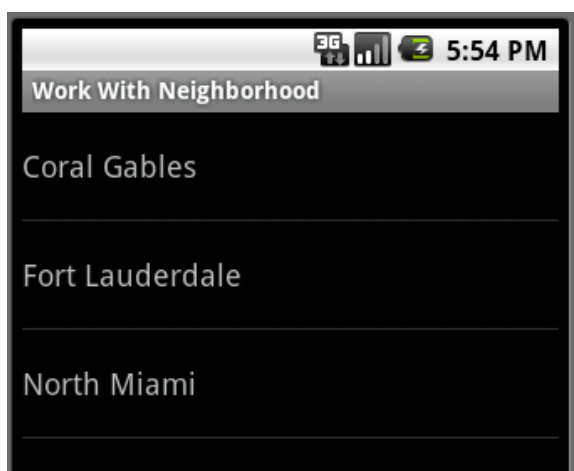
Podemos ver que el nodo List corresponde a esta pantalla en la cuál dentro de un grid tenemos los atributos del neighborhood del vecindario que queremos mostrar.



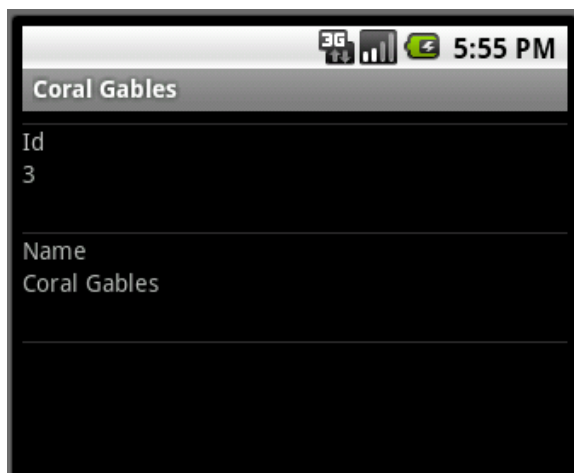
Así en ejecución tenemos: (aquí estamos viendo el dashboard):...



Tenemos aquí el listado de los vecindarios.

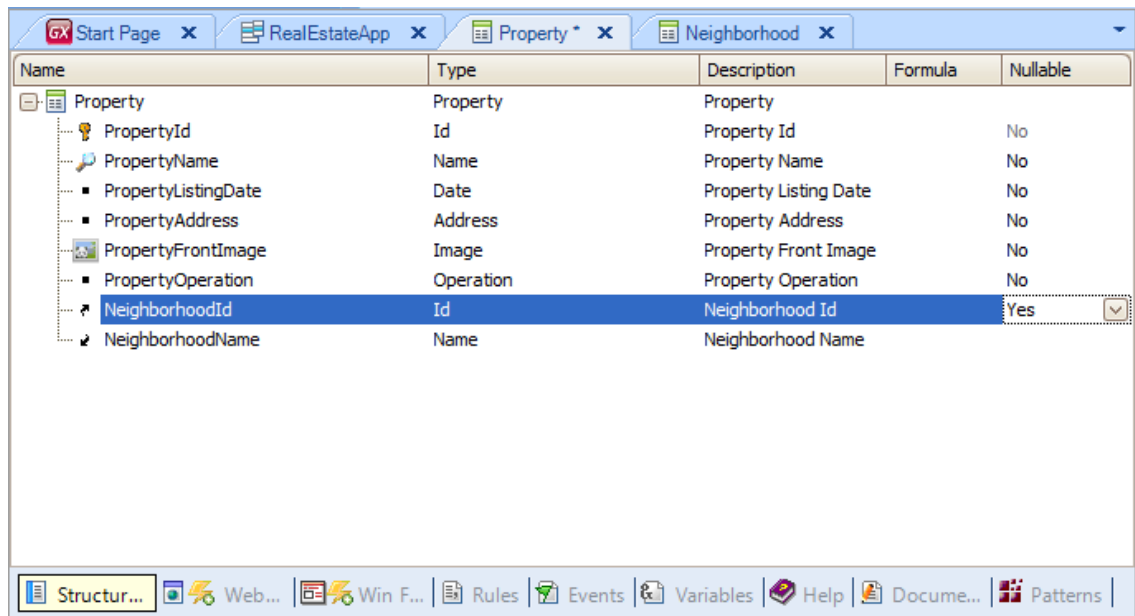


Luego eligiendo uno, vemos la información detallada del mismo, cuyo correlato se encuentra en la sección general.

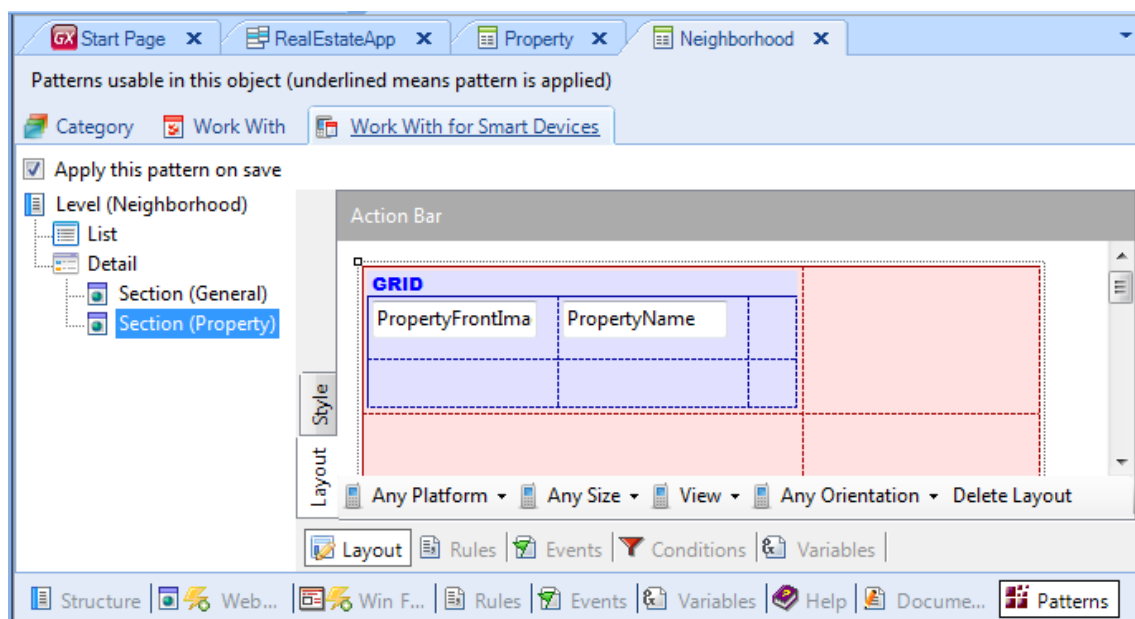


Las propiedades son independientes. Si observamos el detalle de una de ellas vemos su información general sin ninguna vinculación con el vecindario. Ahora establezcamos la relación.

Vamos a la estructura de propiedades y agregamos a NeighborhoodId... pidiéndole que acepte nulos... y a NeighborHoodName. Grabamos.

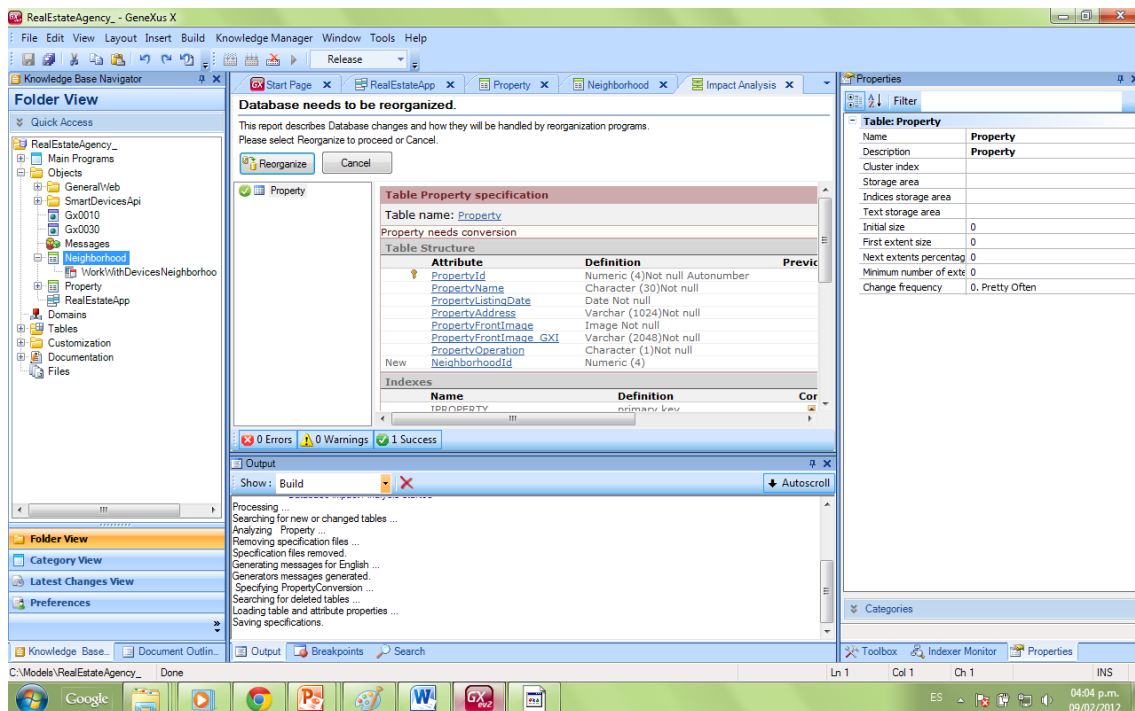


Y veamos qué sucedió con el WorkWith de neighborhood. Está apareciendo una nueva sección. Como vemos está mostrando un grid con las propiedades inmobiliarias. ¿Cuáles? Las de ese vecindario elegido.

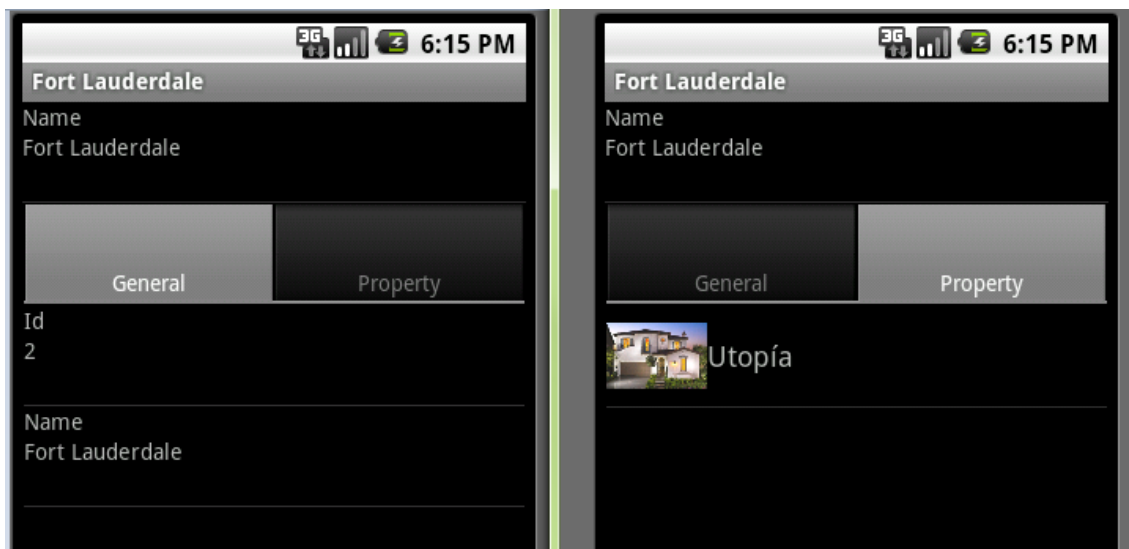


Veámoslo en ejecución. F5.

Reorganicemos.

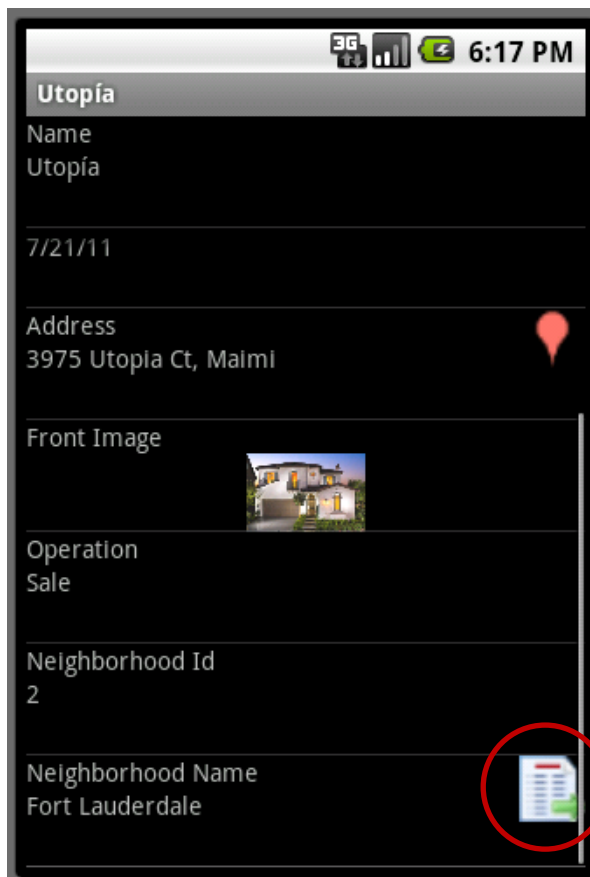


Luego, una vez asignados los vecindarios a las propiedades que teníamos ingresadas, vayamos a ver la lista de vecindarios y elijamos ver el detalle de uno de ellos.

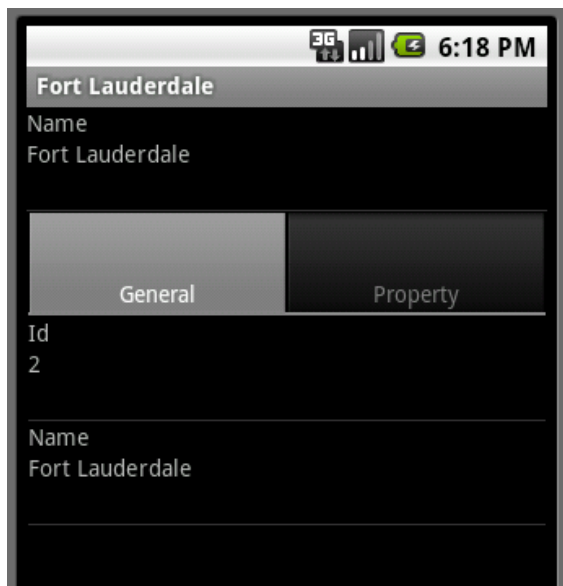


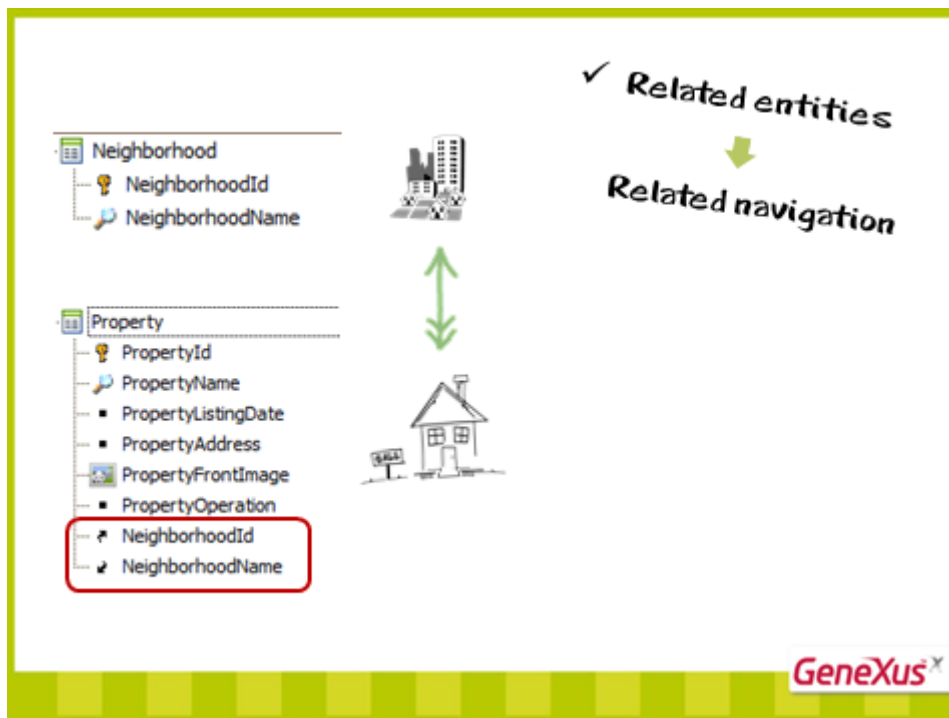
Podemos ver que aparecen dos tabs que coinciden con las secciones que vimos del pattern. El primer tab corresponde a la Section General que contiene la información general del vecindario y el segundo tab corresponde a la otra sección que muestra las propiedades inmobiliarias de ese vecindario. Veamos qué ocurre si elegimos una de ellas...





Nos está llevando al detalle del work with correspondiente a las propiedades, mostrándonos el detalle de esa propiedad y entre su información podemos ver el vecindario, que si seguimos este ícono, nos llevará, cíclicamente, nuevamente, a la información general del vecindario.





Como pudimos ver, si las entidades están relacionadas, las navegaciones ofrecidas también se interrelacionan, sin necesidad de hacer nada.



Con esto hemos visto una pequeña muestra de lo simple y poderoso que es el modelo conceptual.



Qué sigue?

Desea ver cómo personalizar el work with para darle más potencia y adecuarlo a sus necesidades?

Desea ver cómo crear paneles que le permitan extraer y presentar datos de forma flexible, pedir datos al usuario, etc.?

Desea descubrir la arquitectura subyacente?

Continuará...

---

Be smart... Be **GeneXus**

**GeneXus**<sup>®</sup>