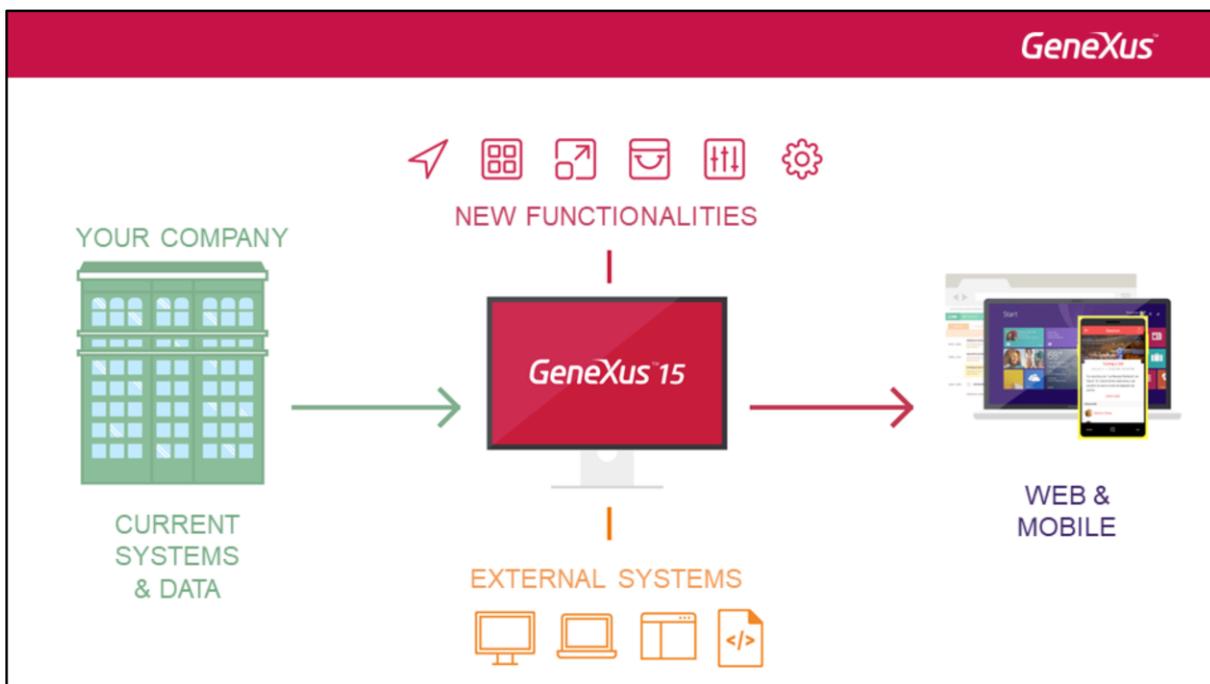


Upgrade course
from GeneXus evolution 3 to GeneXus 15

GeneXus™ 15

GeneXus 15 Overview

Ahora veremos una breve introducción de la versión 15 de GeneXus, actualizada al upgrade 10, para luego, a lo largo del curso, ir entrando y profundizando en los diversos aspectos que aquí se presentan. Por cuestiones de tiempo, algunos temas simplemente se mencionarán y ustedes podrán investigarlos por su cuenta en nuestro wiki.



La realidad a la que nos enfrentamos hoy en día es una en la que las compañías tienen implantados sistemas de software que hacen uso de sus propios datos, pero que no pueden trabajar en forma aislada, sino que necesitan integrarse con sistemas externos, así como ampliar las funcionalidades que proveen de acuerdo a los avances tecnológicos, y con un enfoque fuertemente centrado en la experiencia de usuario. Las soluciones requeridas hoy son aplicaciones web y mobile con una atractiva User Interface (UI). GeneXus 15 se enmarca en esta actualidad.

Digital
Transformation

Integration

User centered
Development

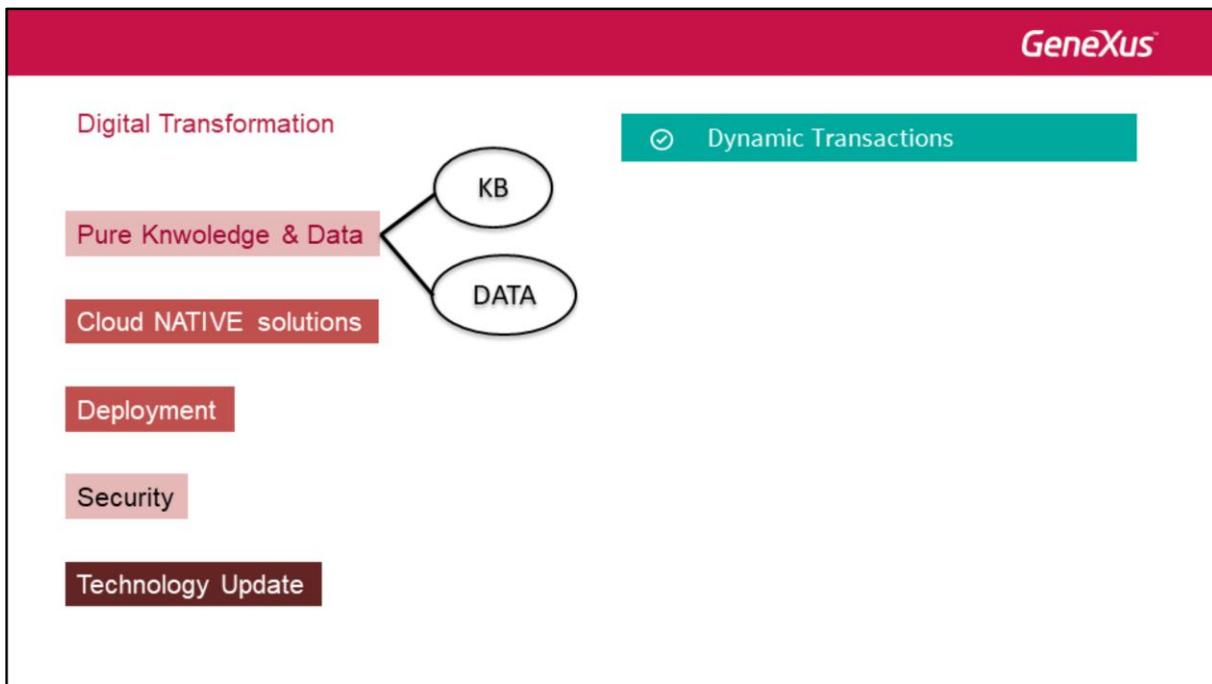
Incremental
Development

Fomenta la transformación digital, permite integrar los sistemas propios con otros, desarrollar la aplicación web y mobile enfocándose en el usuario, y todo a través de un desarrollo incremental mejorado respecto a versiones anteriores.

Veamos brevemente estos pilares para luego pasar a abordar cada cosa en los capítulos siguientes de este curso.

Digital Transformation

Empecemos por la transformación digital. Facilitar la transformación digital y fortalecer los activos digitales es uno de los principales pilares de GeneXus 15.

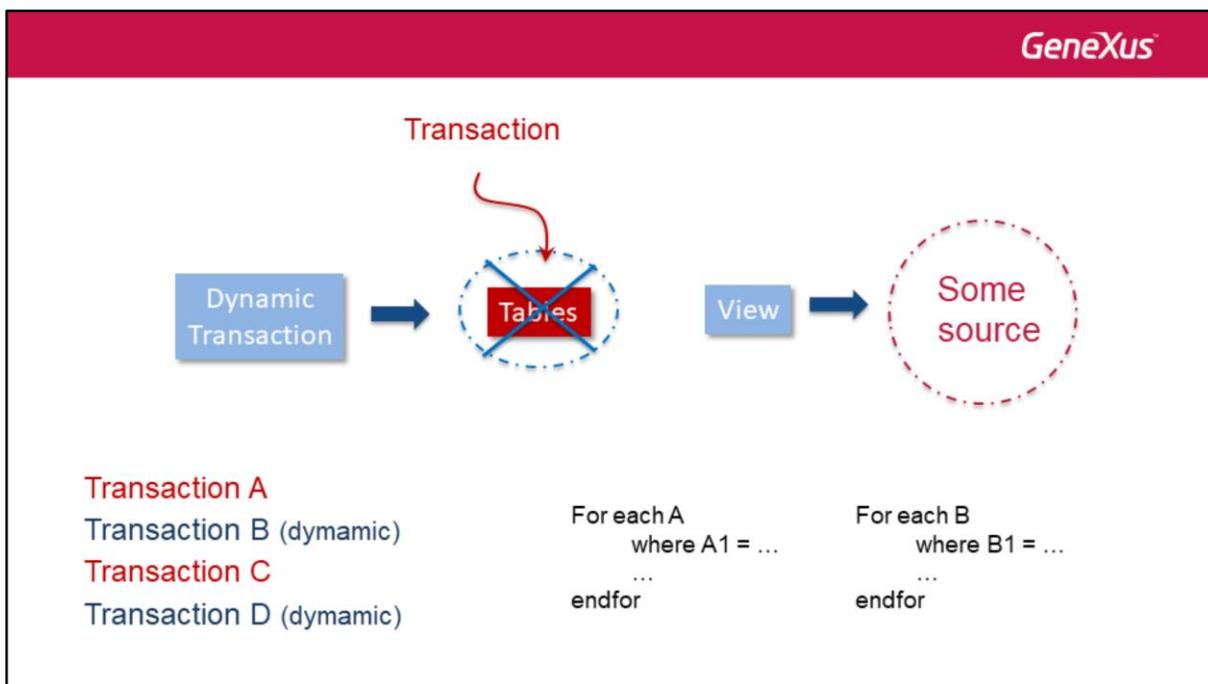


Pueden resumirse en los puntos que vemos.

El primer y muy importante activo digital es la base de conocimiento y los propios datos de la aplicación:

GeneXus 15 facilita la formalización del conocimiento para poder realizar a partir de allí mejores inferencias, y también mejora el manejo de los datos.

A nivel del modelado de datos aparecen las transacciones dinámicas. ¿Qué son?



Transacciones como las que conocemos, solo que en lugar de crear tablas en la base de datos... son como vistas que toman sus datos de diversas fuentes... y permiten así abstraer los detalles físicos y trabajar a alto nivel. La principal ventaja es el tratar uniformemente a las entidades de la realidad modelada, con independencia de dónde se encuentren sus datos y así se logra un altísimo nivel de conceptualización de la realidad, permitiendo que el desarrollador utilice los atributos de la manera estándar (con comando for each, etc.).

Digital Transformation

Pure Knowledge & Data

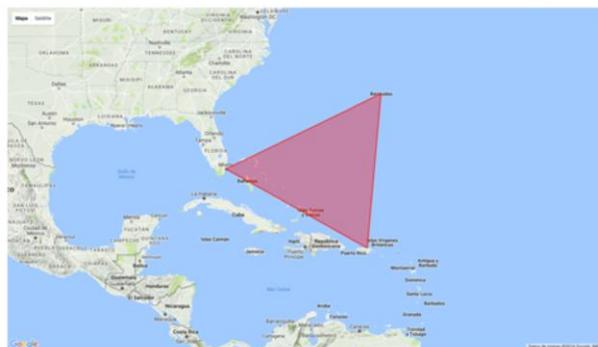
KB

DATA

✔ Dynamic Transactions

✔ Automatic Data Initialization

✔ Geography Data Type



Por otro lado se agrega a las transacciones estándar la posibilidad de declarar en tiempo de diseño los datos con los que queremos que esas transacciones sean inicializadas, sin tener que escribir procedimientos especiales para ello, y pensar desde dónde invocarlos. Todo eso se hace en forma automática con un Data provider que se asocia automáticamente a la transacción, lo que depura la KB, la hace más simple y GeneXus sabe cuándo ejecutarlo.

Por otro lado, cada vez más sistemas almacenan datos geográficos, por lo que los DBMSs actuales proveen soporte nativo tanto para almacenar datos de este tipo como para hacer cálculos complejos que los involucran. Para usar toda su potencia, GeneXus incorpora el tipo de datos Geography que permite almacenar Puntos, Líneas y Polígonos, y operar con ellos. Así, con un simple for each podremos obtener los comercios más cercanos a mi ubicación, o el polígono que define al vecindario donde me encuentro. Esto, dependiendo de la completitud de la versión de GeneXus o del DBMS, terminará supliendo al dominio semántico Geolocation, que se almacenaba la información geográfica en un string en la base de datos.

Digital Transformation

Pure Knowledge & Data

KB

DATA

 Dynamic Transactions

 Automatic Data Initialization

 Geography Data Type

 Internet Of Things (IoT)

Major Components of IoT



Por otro lado, asociado al tema de la transformación digital no podemos dejar de nombrar un escenario que se está tornando muy importante: la posibilidad de recibir datos de distintos tipos de sensores y procesarlos, definiendo reglas que permitan establecer acciones a tomar en base a las medidas registradas, como enviar notificaciones con alarmas a las personas asignadas. Todo esto a través de un servidor de IoT con servicios rest, con un backend que ofrezca paneles para graficar la información con líneas de tiempo, entre otras cosas.

Existen hoy distintos servidores de IoT en la nube, que ofrecen la posibilidad de manejar la información de una cantidad enorme de sensores, escalable. Por ejemplo, IoT Hub de Microsoft Azure.

Pero en GeneXus...

The screenshot shows a web browser window displaying the GeneXus Wiki page for IoT Server. The page has a red header with the GeneXus logo and navigation links. A table on the left side of the page provides links for KnowledgeBase, GeneXus Server URL, and RSS URL, all pointing to IoTServer and here. The main content area is titled 'IoT SERVER' and includes a list of links for Knowledge Base, Backend, How to Get Started, How to Move Forward, Licensing Information, and How to Collaborate. The right side of the page features the title '<IoT Server' and a description of the Knowledge Base, stating that it allows users to quickly begin recording sensor data and processing it for decision making. The description also mentions that sensor data is received through a REST API and that the information is received in a specific format, including an API key, sensor values, and a time stamp. Alarms can be defined with triggering rules and people assigned to them.

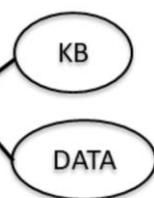
Pero en GeneXus hemos construido una KB open source, IoTServer (aquí, en nuestro wiki, pueden ver toda la documentación).

En el backend ofrecido en esa KB se define la metadata (que será la información de los sensores y las medidas que éstos entregarán) y las reglas que establecen qué debe hacerse frente a determinadas situaciones, como enviar notificaciones, por ejemplo. Los datos de los sensores son recibidos vía una API REST implementada en la KB. Para un número no demasiado grande de sensores esta solución funciona bien. El backend incluye pantallas con dashboards que grafican con timelines los valores recibidos, entre otras cosas.

Como ejemplo, un proyecto en marcha es hacer un mapa sonoro de Montevideo. ¿Qué significa esto? Tener sensores que midan los decibeles en distintos vecindarios de la ciudad, y envíen esos datos ya sea a gateways que los enviarán como input a servicios, o directamente a esos servicios (apis).

Digital Transformation

Pure Knowledge & Data

 Dynamic Transactions Automatic Data Initialization Geography Data Type Internet Of Things (IOT) Artificial Intelligence (AI)

Y otro campo de aplicación que se está tornando omnipresente hoy es la Inteligencia Artificial, por ejemplo para el reconocimiento de imágenes, o reconocimiento de texto en lenguaje natural, que permita realizar acciones como por ejemplo implementar un Chatbot, o brindar respuestas automáticas a preguntas en lenguaje natural.

Todo lo que se conoce como Machine Learning y Deep Learning. Hoy varias plataformas en la nube ofrecen Servicios Cognitivos de este tipo, análisis predictivos y varios más, en la forma de APIs REST. Por ejemplo, tenemos Watson de IBM, las apis de servicios cognitivos de Microsoft, o SAP Leonardo, entre otros.

Volveremos brevemente sobre esto en un rato, cuando abordemos el pilar de la Integración. A quien le interese adentrarse en estos temas, le recomendamos tal vez empezar por ver las charlas sobre Watson y sobre Inteligencia Artificial que se dieron en los eventos 26 (cuando empezamos a hablar de todo esto) y 27, es decir, los de los años 2016 y 2017, y por supuesto, las que se darán este año, 2018.

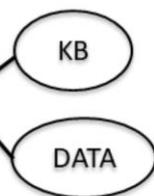
www.genexus.com/meeting2016

www.genexus.com/meeting2017

Digital Transformation

Pure Knowledge & Data

Cloud NATIVE solutions



✔ Dynamic Transactions

✔ Automatic Data Initialization

✔ Geography Data Type

✔ Internet Of Things (IOT)

✔ Artificial Intelligence (AI)

✔ SAP Hana DB

Otra novedad en cuanto a datos es el soporte del DBMS SAP HANA. Se trata de un DBMS que maneja todo en memoria, utiliza las ventajas del procesamiento en paralelo y además permite guardar tablas por filas o columnas. Así, no sólo es altamente performante, sino también altamente escalable.

Pero la gran ventaja es que es el DBMS promovido por SAP en su nueva línea de productos y servicios en SAP Cloud Platform. Una mención especial es el hecho de que sumado a esto se ha creado una especialización de GeneXus, denominada GeneXus for SAP systems, para aquellos usuarios del mundo SAP que quieran desarrollar aplicaciones para esa plataforma particular. Desde la versión estándar de GeneXus también podremos hacerlo, con el soporte de esta DB Hana.

Esto nos conduce al siguiente tema que es el desarrollo de soluciones nativas para Cloud.

Digital Transformation

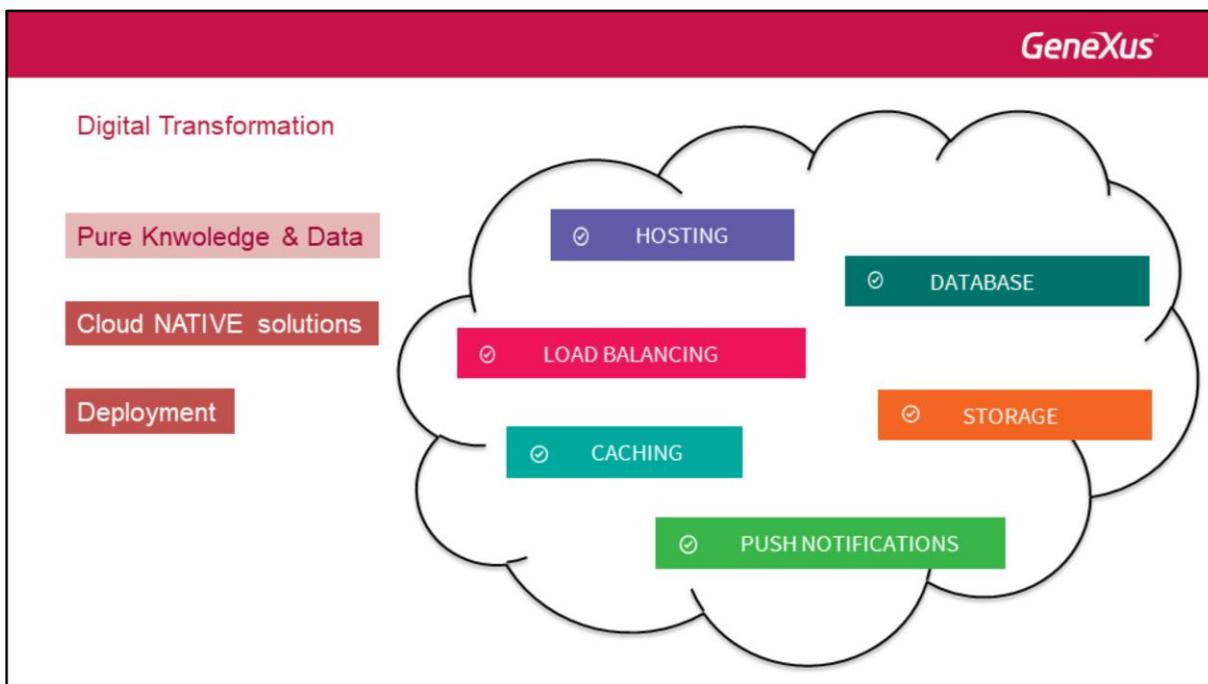
Pure Knowledge & Data

Cloud NATIVE solutions



Los proveedores de Cloud están realizando enormes esfuerzos dirigidos hacia la transformación digital, ofreciendo sus plataformas como servicio y dictando normas acerca de cómo debemos desarrollar nuestros sistemas para que funcionen correctamente y escalen en sus nubes.

GeneXus se suma a esta tendencia, respetando estas reglas, para que nuestras aplicaciones puedan funcionar de manera **nativa** en todas estas plataformas, realzando la capacidad cross-platform de GeneXus.



El hecho de que nuestras apps sean Cloud nativas significa que además de funcionar en esas plataformas, es decir, de ser hosteadas allí, aprovechan sus funcionalidades y servicios.

Por ejemplo nuestros sistemas se pueden integrar a los servicios de Bases de datos que estos proveedores ofrecen, no solamente los servicios de las bases de datos tradicionales que conocemos con GeneXus sino otros que están diseñados especialmente para escalar en la nube, como Aurora para Amazon.

Relacionado con la escalabilidad, las aplicaciones web GeneXus ahora escalan mejor en forma flexible con Load Balancing.

Permiten utilizar los servicios de Storage, es decir, espacios de almacenamiento no relacionales donde poder almacenar imágenes y demás archivos multimedia, de forma tal de aliviar a nuestro DBMS, almacenando en él solamente links a los archivos almacenados en esos servicios de Storage, beneficiando la escalabilidad, la baja en los costos de tráfico y la mejora de performance, ya que los proveedores tienen mecanismos para enviar esos archivos a los usuarios finales.

Además proveen servicios de caching distribuido, y estos son soportados en GeneXus 15, también a través de una simple configuración. Esto mejora la escalabilidad enormemente.

Por último, las aplicaciones requieren notificar continuamente a sus usuarios y para ello las nubes ofrecen servicios de Push Notifications que permiten enviar miles y miles de notificaciones por segundo. GeneXus se integra con uno especial denominado OneSignal.

Pero además, estos proveedores de Cloud indican cómo debemos publicar estas soluciones.

Digital Transformation

Pure Knowledge & Data

Cloud NATIVE solutions

Deployment

⊙ Packaging

⊙ Deployment to Cloud (PaaS)

⊙ GeneXus Cloud Deployment Services

GeneXus 15 facilita el Deployment de varias maneras.

Por un lado, ofrece un mecanismo muy sencillo para empaquetar la app y subirla a un servidor web, se trate de plataforma Java o .Net.

Y en caso de querer subir la solución a Cloud, también publica los binarios y recursos automáticamente en la nube, en cualquiera de los proveedores con los que GeneXus trabaja (agregando el desarrollador las credenciales necesarias).

Y como instalar no es únicamente subir los binarios (y recursos) sino también es ejecutar reorganizaciones, inicializar datos, cuidar la seguridad y luego de instalar también es gestionar las publicaciones, creamos GeneXus Cloud, un conjunto de servicios y tecnologías. Es una solución completa, es decir, resuelve todo el ciclo. Con apenas un clic sube toda la aplicación, se ejecuta la reorganización, la inicialización de la base de datos, y luego ayuda a gestionar las versiones en producción.

Digital Transformation

Pure Knowledge & Data

Cloud NATIVE solutions

Deployment

Security

☑ GeneXus Access Manager (GAM)

☑ Server-side

☑ Client-side

Otro de los aspectos importantes es la seguridad.

Se ha mejorado el GAM, por ejemplo en cuanto al manejo interno de tokens, en apego a los estándares y guías relacionadas.

Se han fortalecido los componentes que ejecutan del lado del server y los que corren del lado del cliente. Y también se ha minimizado el tráfico que, además de impactar en la performance, también lo hace en la seguridad.

Digital Transformation

Pure Knowledge & Data

Cloud NATIVE solutions

Deployment

Security

Technology Update



Apple recomienda el nuevo lenguaje Swift para desarrollo de aplicaciones móviles nativas para iOS, en lugar de Objective-C, razón por la cual GeneXus ahora genera sus aplicaciones iOS en este lenguaje.

En los primeros upgrades de esta versión se soportaba iOS 10, pero desde el upgrade 7 se soporta también el 11. Y para Android de soportar el 7.0 se pasa a partir del upgrade 9 a soportar también Android 8. Se incorporando, además, las nuevas guías de diseño de las plataformas.

Todo eso de manera transparente para el desarrollador.

Integration

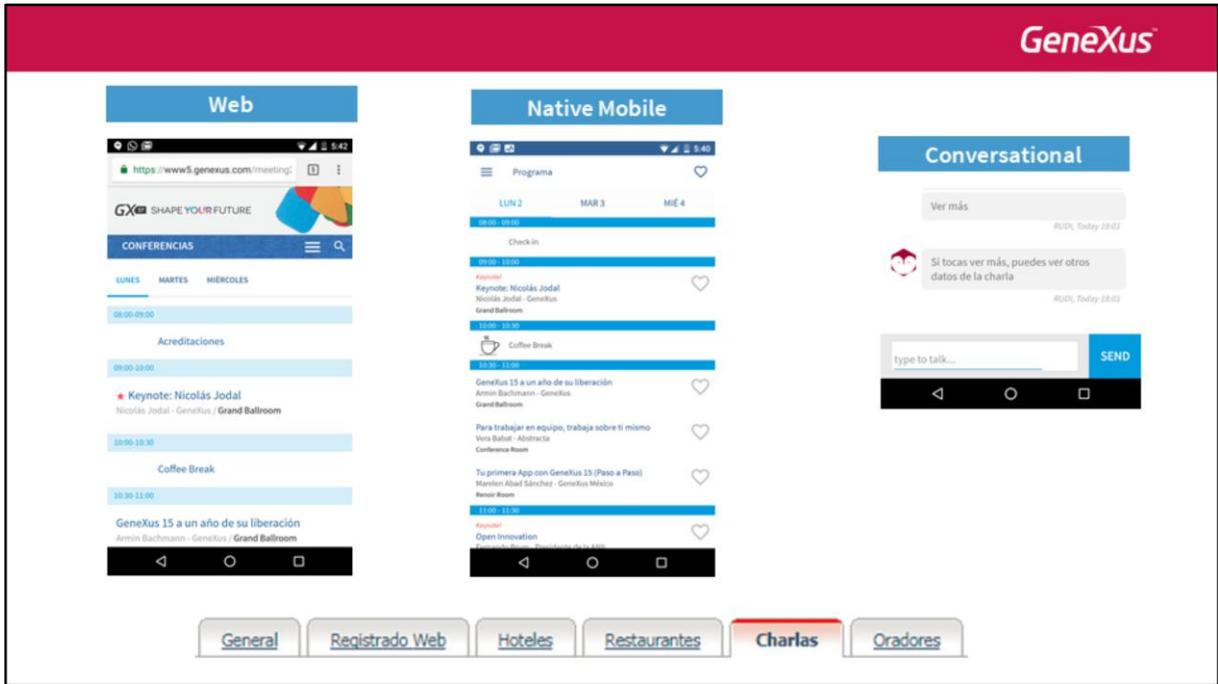
Pasemos al pilar de la Integración.

Integration



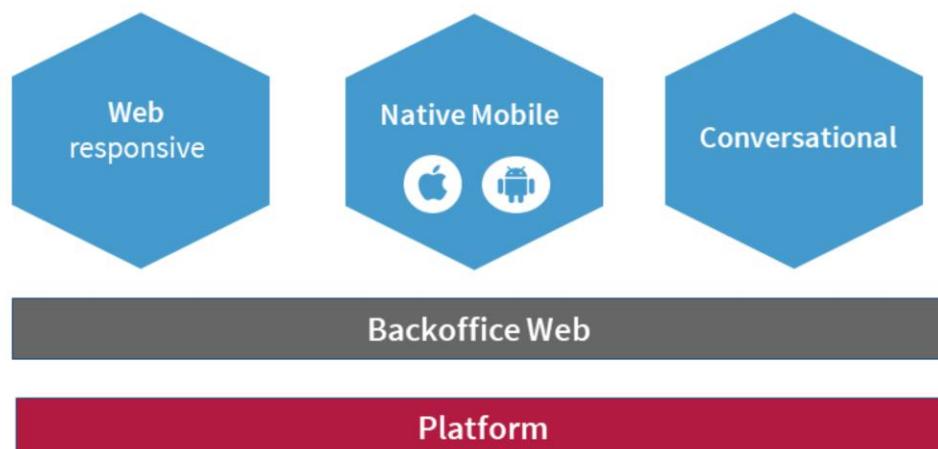
Hoy en día los sistemas desarrollados con GeneXus requieren ofrecer una solución Web responsive, que pueda satisfacer la experiencia de un usuario que utiliza tanto un desktop como un dispositivo móvil; requiere ofrecer una solución nativa tanto para iOS como para Android; y por supuesto un Backoffice web para mantener los datos.

Además, se van incorporando otros tipos de componentes, como uno conversacional, a la manera de un Chatbot como el que se implementó para la aplicación del último evento GeneXus...



La app del evento tenía un componente web responsive, para los usuarios que querían ver de qué iba la cosa sin la molestia de tener que descargarse e instalarse la app móvil; luego tenía la app móvil (tanto en Android como en iOS) para los usuarios más comprometidos que sí irían al evento, por supuesto un backend web para mantener toda la información, y un chatbot dónde en lenguaje natural se le podían hacer un montón de preguntas a Rudi que ella respondía en la medida en que había sido entrenada para ello.

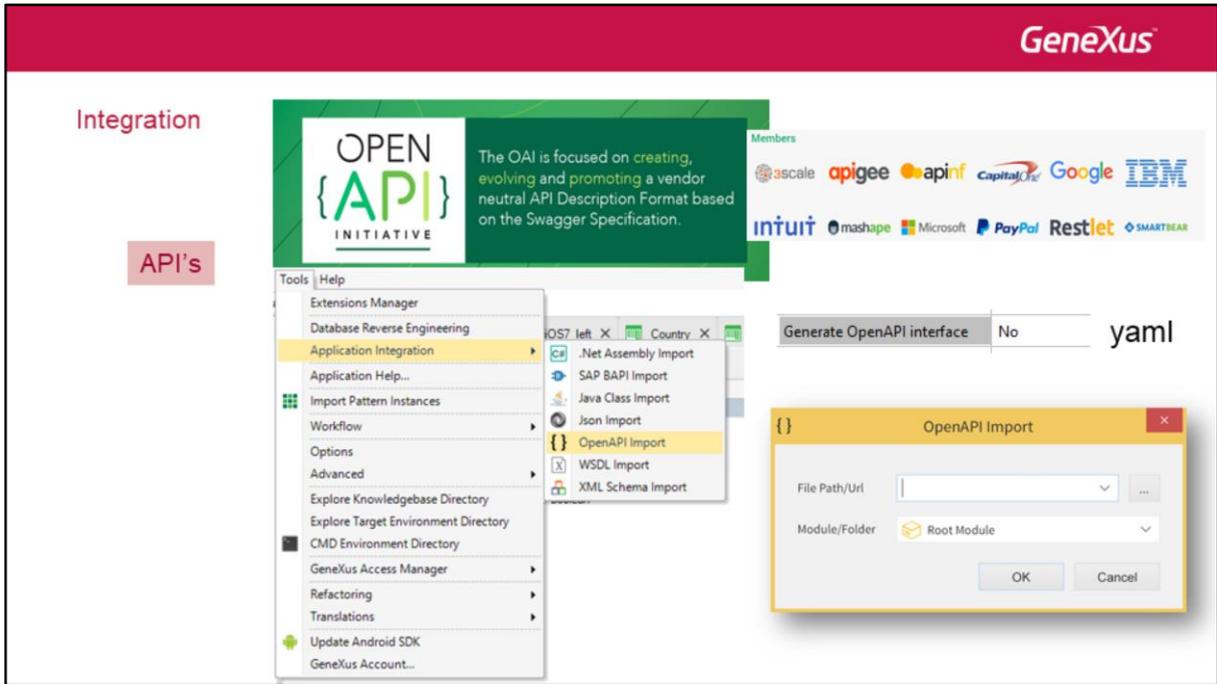
Integration



Este componente conversacional se implementó utilizando los servicios de Machine Learning relacionados con la interpretación de textos en lenguaje natural de Watson, y en la próxima versión de GeneXus estará incorporado como **el generador ChatBot**, que no solo funcionará con los servicios de inteligencia artificial provistos por Watson, sino con los de Amazon, Google y demás – con los proveedores que vayan apareciendo.

En definitiva, este componente de una app GeneXus ya no puede trabajar en forma aislada, sino que tiene que integrarse a servicios provistos por plataformas. Y hay muchos otros ejemplos, cada vez más, de aplicaciones que deben integrarse con servicios provistos por terceros, por ejemplo por Bantotal para aplicaciones financieras, integración con ERPs, con MercadoLibre, o, como ya dijimos, con servicios de inteligencia artificial o servicios para internet de las cosas.

Esta integración se da en su mayoría a través de APIs.



Y hoy una enorme cantidad de esas APIs están expuestas como servicios REST, descriptos para su utilización por terceros siguiendo el estándar Open API.

En GeneXus 15 se proveen dos funcionalidades:

La primera, el poder exponer procedimientos, data providers o business components como servicios rest documentados siguiendo este estándar: Open API, para que otras aplicaciones puedan consumirlos fácilmente, a partir de esa documentación.

Para esto se incorporó la propiedad **Generate OpenAPI interface** para generar los archivos yaml de documentación de las apis rest que se creen (a partir de procs, data providers y business components). El valor default es No.

Por ejemplo, en la KB IoTServer de la que hablamos antes, creada con GeneXus y abierta para ser descargada por cualquiera, se necesita exponer una API para cargar datos desde los sensores. Esa API se expone como API rest, generando la documentación a partir de la propiedad Generate OpenAPI interface (pasándola a Yes).

Y la segunda funcionalidad es el poder importar API's provistas por otros, documentadas según OpenAPI. Esta documentación le permite a GeneXus saber qué objetos (externos, procs, dominios, tipos de datos, etc.) crear para poder utilizar la API siguiendo el lenguaje GeneXus de modo de abstraer toda la complejidad.

Un ejemplo donde se importa una API en GeneXus que tiene un archivo de documentación que sigue los estándares de OpenAPI es una KB que se está trabajando para reconocer la composición de una pila de chatarra. Para ello se utilizan servicios (apis) de inteligencia artificial provistas por SAP Leonardo, por lo que esas apis deben importarse en GeneXus para que el desarrollador pueda utilizarlas como Objetos externos con SDTs y todo lo demás que ya conocemos.

Integration

Modules

 Sharing Knowledge

 Preserve Privacy

 Simplify Understanding

Module API +



e.g.

GeneXus SDK for MercadoLibre

GeneXus SDK for Watson

EOs, SDTs,
domains

Otro aspecto ciertamente relacionado que hace a la integración es la posibilidad de desarrollar y compartir módulos GeneXus, permitiendo compartir el conocimiento entre KBs, cuidando la privacidad del código y simplificando la comprensión de nuestras bases de conocimiento. ¿Qué es lo que se comparte, o viaja de una KB a otra? Las APIs y los binarios que constituyen el modulo (no los objetos GeneXus que implementan sus funcionalidades).

Lo veremos luego en detalle, pero ya podemos presentar como un ejemplo de esto el SDK de MercadoLibre, o el SDK de Watson, que por el momento son módulos que se importan en la KB, para poder utilizar esas APIs como objetos externos GeneXus, que nos permitan ejecutar los servicios a la manera GeneXus, ocultando toda la complejidad. Como mencionaremos luego, esto por el momento se importa en forma explícita. Más adelante estará incorporado por default en el propio módulo GeneXus que viene con cada KB, del que ya hablaremos en un rato, cuando trabajemos este tema Módulos.

User Centered Development

(UX + DX)

Ahora pasemos a un pilar fundamental: el desarrollo centrado en el usuario. Estamos hablando fundamentalmente de la User Experience, pero también de la experiencia del desarrollador.

User centered development



✔ Great UX by Default



✔ Flexibility for UX



✔ UX Modeling

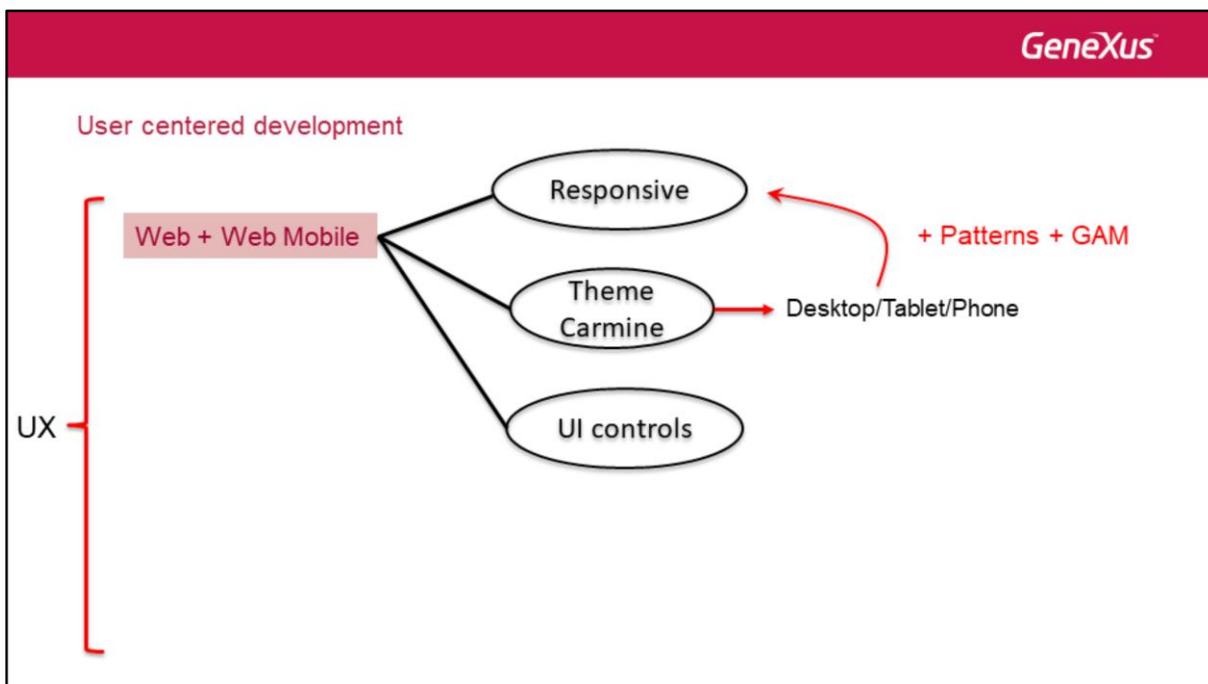
Nos centramos en estos tres aspectos fundamentales: que las apps web y móviles consigan una excelente experiencia de usuario en forma predeterminada. Por ejemplo, para el caso de apps web, que sean totalmente responsivas por defecto.

Lograr flexibilidad, potencia y facilidades en el modelado.

User centered development



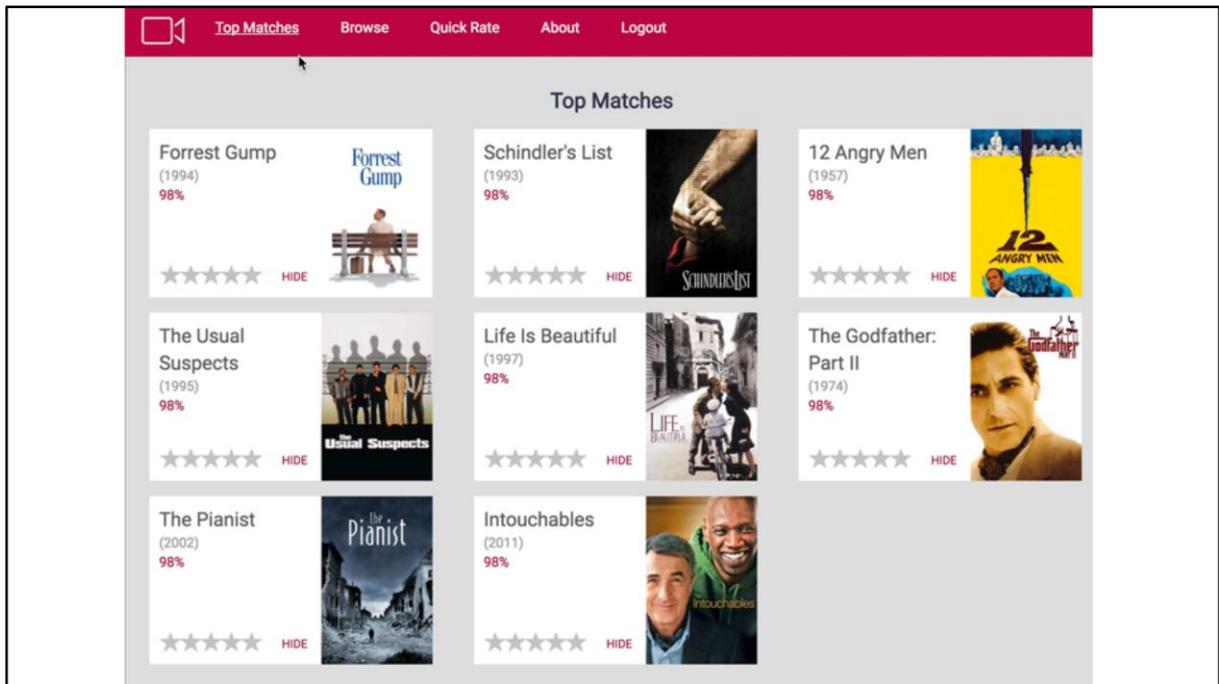
Aquí vemos los principales puntos divididos en experiencia de usuario y experiencia del desarrollador.



Para el desarrollo de apps web y web mobile, se ha modificado el default, y ahora, si no se indica lo contrario, la app será Responsive.

Se ha creado un nuevo theme, Carmine, el theme default, que determina el diseño de la app tanto para Desktop (sea touch o no), como para Tablet o phone. Es decir, toma en cuenta la responsividad, y aplica tanto para los Patterns como para GAM.

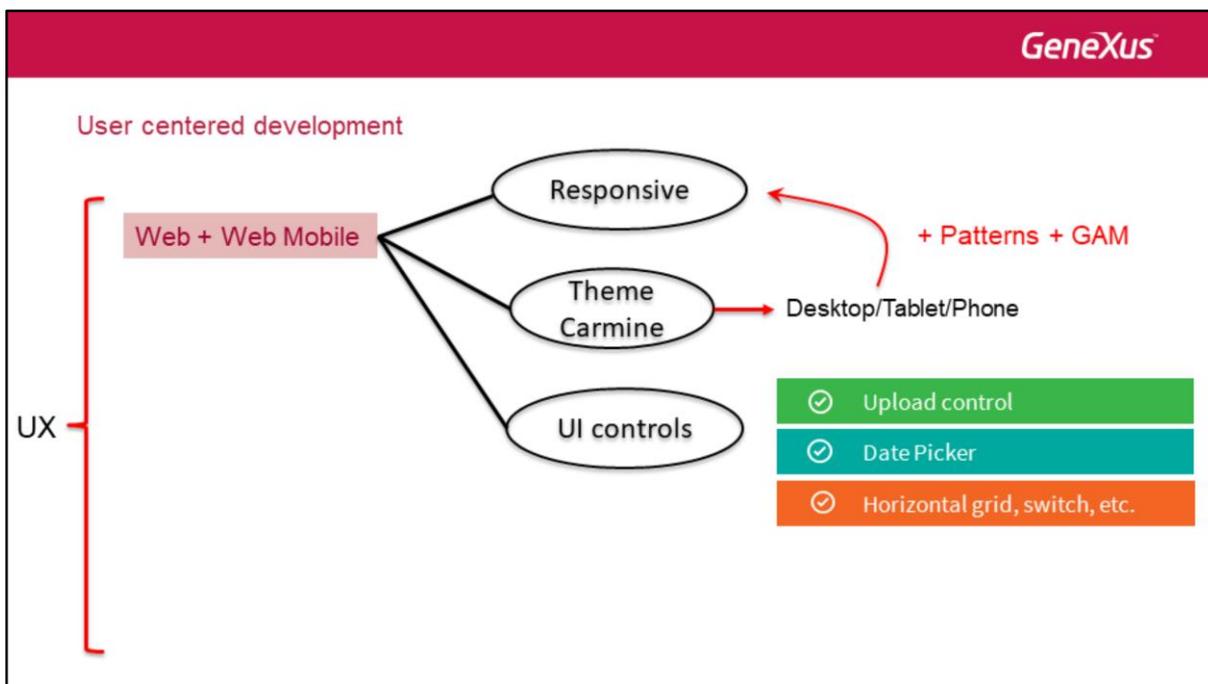
Por otro lado se han creado controles nuevos para los forms o layouts, que puedan funcionar de manera responsive...



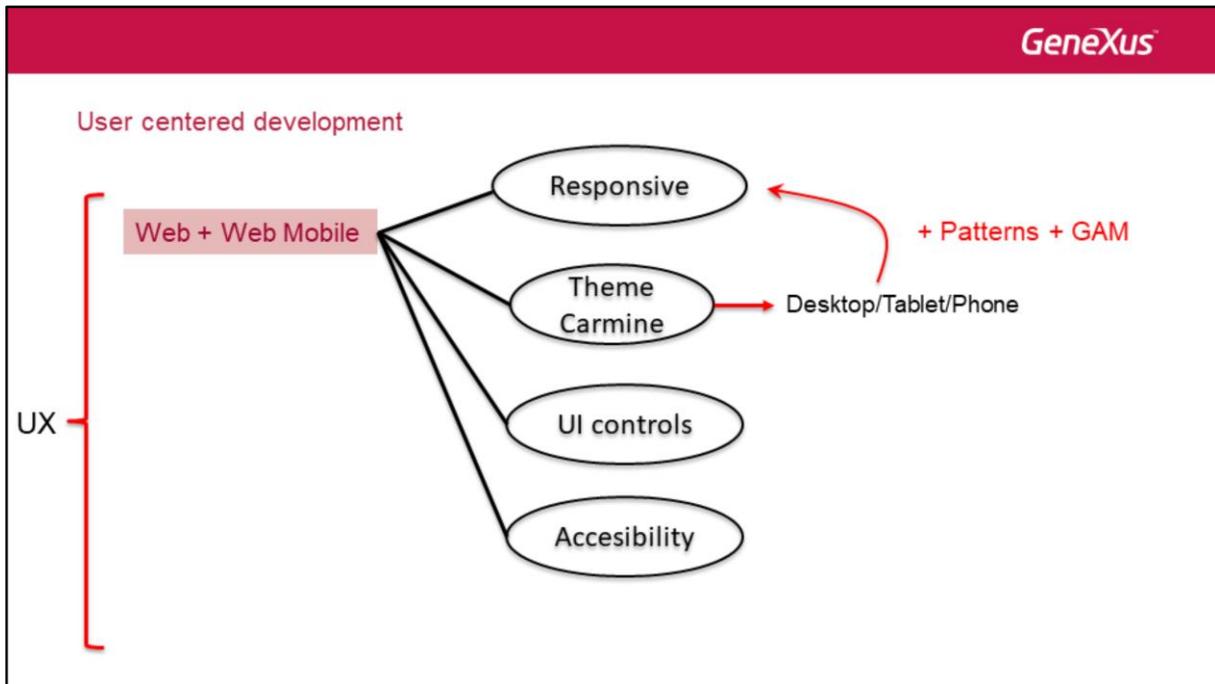
Como podemos ver aquí, donde tenemos una app web con una grilla que muestra películas, y arriba un control action group de tipo menú. Vemos que al achicar la pantalla los controles se van adaptando, el grid pasa de mostrar tres columnas a mostrar una sola y el menú llega a transformarse en el típico menú hamburguesa.

Vemos las opciones que muestra...

Si ahora ampliamos la pantalla, vuelve a presentar todas las opciones en forma horizontal y más columnas para el grid.

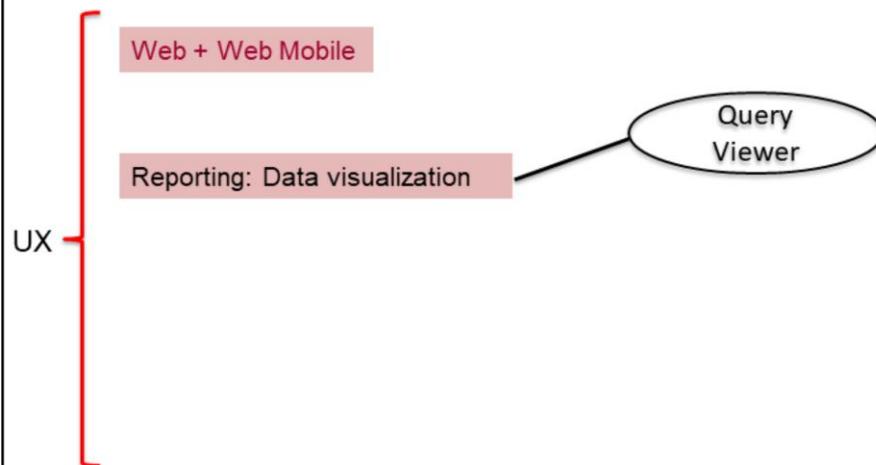


Además de haber agregado nuevos controles, se ha mejorado la flexibilidad de otros. Algunos ejemplos son el Upload control para subir archivos, el Date Picker (que al ejecutarse una app mobile web tanto en iOS como en Android utiliza ahora los controles nativos para permitir seleccionar una fecha), y otros.

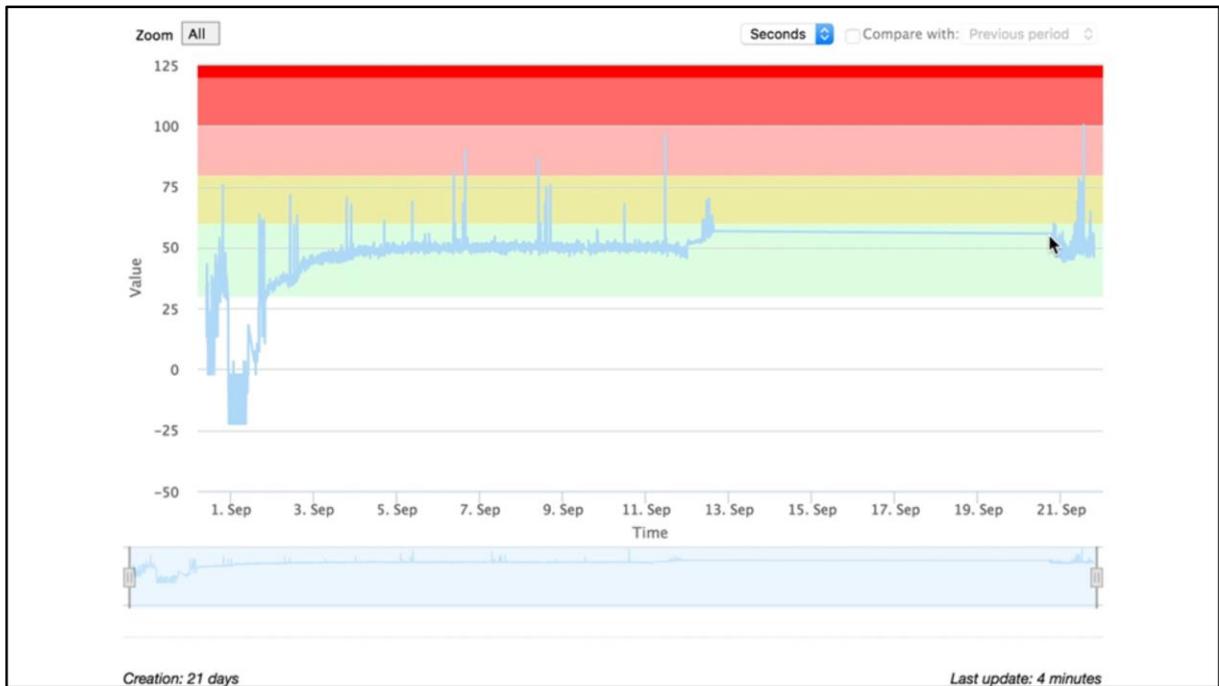


Y se ha realizado un trabajo interesante para hacer las apps accesibles para usuarios con discapacidades. Hay aspectos que infiere automáticamente GeneXus, y otros que los desarrolladores deben configurar explícitamente.

User centered development



A nivel de Reporting, se ha mejorado el Query viewer, especialmente en lo que hace a las gráficas y a la performance, y su UI es personalizable a través del theme.



Cada vez más se necesita graficar variaciones de alguna variable en el tiempo, fundamentalmente tomada de sensores. Por ejemplo, aquí vemos las variaciones tomadas por un sensor de sonido, de decibeles en el tiempo. Las gráficas timeline nos permiten analizar los datos, ver en qué horas se dieron picos, por ejemplo, y aumentar la precisión de la información para el rango elegido.

Están siendo muy utilizadas en dashboards.

Reporting: Data visualization



≈ 44,741.25
(Invoice Total Amount)

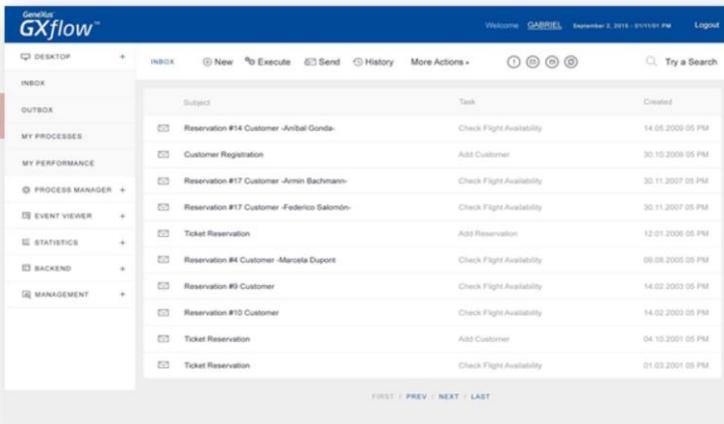
Junto con otras gráficas como la que vemos, que muestra las ventas de distintos granos por región.

Para los dashboards no solamente vemos el incremento de estos tipos de gráficas, sino también la utilización de cards, como la que vemos aquí, que muestra la tendencia de cierto indicador (en este caso muestra que la facturación está creciendo).

User centered development

UX

- Web + Web Mobile
- Reporting: Data visualization
- Workflow based systems



The screenshot displays the Gxflow user interface. The top navigation bar includes the Gxflow logo, a user profile (GABRIEL), the date (September 2, 2018), and the time (11:11:01 PM). Below the navigation bar, there is a sidebar menu with options like DESKTOP, INBOX, OUTBOX, MY PROCESSES, MY PERFORMANCE, PROCESS MANAGER, EVENT VIEWER, STATISTICS, BACKEND, and MANAGEMENT. The main content area shows an inbox with a table of tasks. The table has columns for Subject, Task, and Created. The tasks listed include Reservation #14, Customer Registration, Reservation #17, Reservation #17, Ticket Reservation, Reservation #4, Reservation #9, Reservation #10, and Ticket Reservation.

Subject	Task	Created
Reservation #14 Customer -Anibal Gonda-	Check Flight Availability	14.05.2009 05 PM
Customer Registration	Add Customer	30.10.2009 05 PM
Reservation #17 Customer -Armin Bachmann-	Check Flight Availability	30.11.2007 05 PM
Reservation #17 Customer -Federico Salomón-	Check Flight Availability	30.11.2007 05 PM
Ticket Reservation	Add Reservation	12.01.2006 05 PM
Reservation #4 Customer -Marcela Dupont	Check Flight Availability	09.08.2005 05 PM
Reservation #9 Customer	Check Flight Availability	14.02.2003 05 PM
Reservation #10 Customer	Check Flight Availability	14.02.2003 05 PM
Ticket Reservation	Add Customer	04.10.2001 05 PM
Ticket Reservation	Check Flight Availability	01.03.2001 05 PM

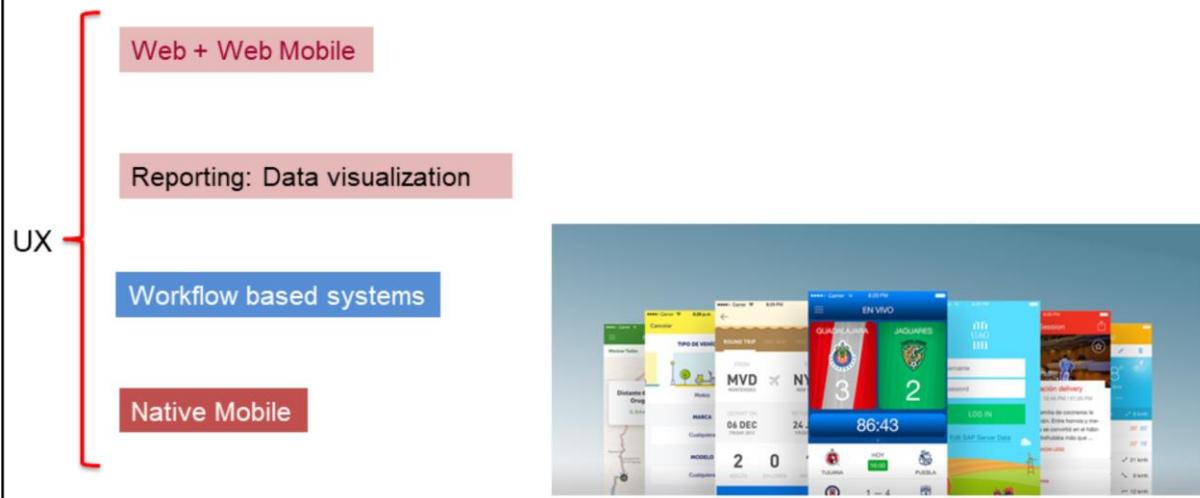
Con respecto a Workflow, la interfaz de usuario del Inbox de Gxflow fue renovada.

Gxflow agrega soporte SAP Hana y Oracle (del 9 al 11).

Cada operación de CRUD sobre usuarios y roles en el backend GAM se ve reflejado automáticamente en cliente de Gxflow y viceversa.

Se permite el Import/Export de usuarios en un cliente Gxflow, etc.

User centered development



También para las apps nativas móviles se han creado nuevos controles, se han extendido las funcionalidades de controles anteriores, y se han agregado efectos visuales de acuerdo al avance tecnológico de estas plataformas, siguiendo las nuevas guías de diseño, tanto de Android como de iOS.

Profundizaremos sobre esto en un capítulo aparte.

User centered development



Respecto a la experiencia del desarrollador... se ha mejorado el IDE en varios aspectos, se ha modificado ligeramente la Start Page, han aparecido algunas mejoras en el lenguaje (por ejemplo: se han agregado cláusulas de paginado para el comando for each, se han extendido los métodos de actualización en Business Components para colecciones de ellos, pueden dispararse eventos de external objects, etc.).

Y se ha agregado una funcionalidad de vital importancia para el desarrollador: Live editing. La veremos dentro del último pilar...

Incremental Development

Además en esta versión de GeneXus se trabajó en mejorar lo que podemos llamar una huella de identidad de GeneXus que es el Desarrollo Incremental.

Incremental Development

Team Development

GeneXus
Server

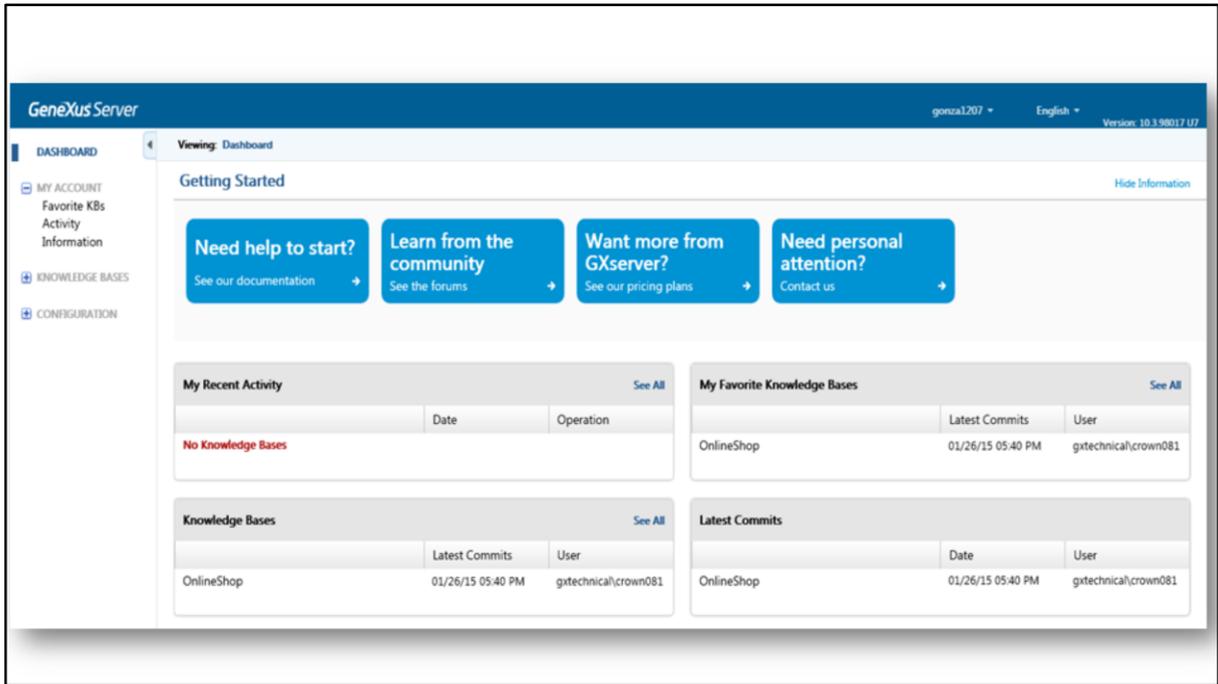
Shorten Prototyping Cycles

Eliminate Prototyping Cycles

Así, se trabajó en acelerar y facilitar el trabajo en equipo, acortar los ciclos de prototipación y en algunos casos ¡eliminarlos!

Presentemos brevemente cada uno.

Para mejorar y facilitar el trabajo en equipo, se han realizado cambios interesantes en GeneXus Server...



...que ahora tiene una nueva Consola.

Incremental Development

Team Development

GeneXus
Server

✓ Changesets

✓ Partial Commit

✓ Blame

Shorten Prototyping Cycles

Eliminate Prototyping Cycles

A partir de ahora se pueden agrupar cambios según funcionalidad, para que no queden todos los cambios de distintas funcionalidades entreverados.

Además se puede hacer commit de una parte de un objeto.

Y con la funcionalidad de Blame se puede saber quién tocó qué línea de código.

Además se puede hacer un Update to a Specific revision, entre algunas otras cosas.

Incremental Development

Team Development

Shorten Prototyping Cycles

Build
Performance

✔ Specification in Background

✔ Code Generation Performance

✔ Compilation Performance

Eliminate Prototyping Cycles

Se han agilizado los procesos de Build, lo que hace a los ciclos de prototipación más rápidos.

Mientras el desarrollador va diseñando, describiendo, GeneXus va especificando en background.

Se han modificado algunas técnicas para la generación de código, que utilizan menos recursos, por lo que el proceso es más rápido.

Y también la compilación es mucho más rápida. Se aprecia claramente en iOS, pero mucho más en Android.

Ahora, podemos mejorar el build, tratando de generar menos o de generar más rápido, pero la mejor forma es

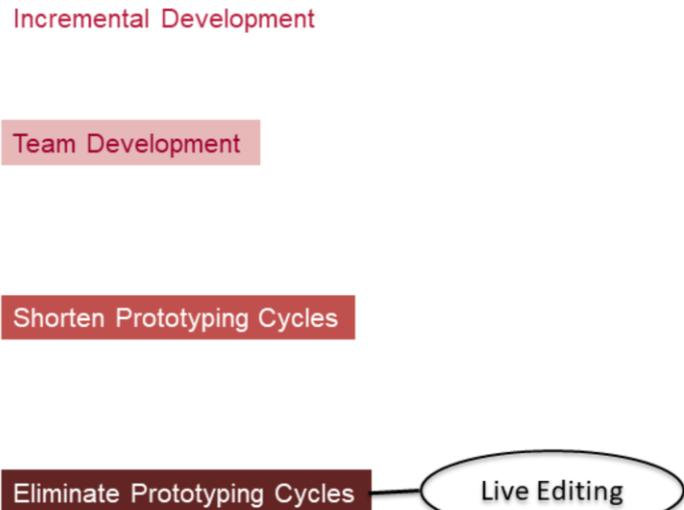
Incremental Development

Team Development

Shorten Prototyping Cycles

Eliminate Prototyping Cycles

Live Editing



la herramienta Live Editing, que nos permite hacer cambios en el diseño y comportamiento de nuestras apps móviles y visualizarlos ¡en vivo!, en el propio dispositivo, sin siquiera haberlos grabado aún. Si nos convencen, entonces los grabamos, y si no, los descartamos sin más. Esto significa que podemos ver al instante un cambio, sin necesidad de un build. Veremos esta herramienta cuando estudiemos lo nuevo en las apps nativas móviles, aunque también puede utilizarse para aplicaciones web.

Next

- Setup improvements & KB conversión
- GeneXus 15 features (among the ones shown here)
- RESOURCES:
 - Training Site: training.genexus.com
 - GeneXus 15 Info: genexus.com/genexus15
 - Technical Documentation: genexus.com/v15
 - Support: genexus.com/issuetracking

En lo que sigue veremos mejoras en el setup y mencionaremos cómo convertir KBs desarrolladas con Evolution 3 a GeneXus 15.

La intención de los capítulos que siguen es presentar cada feature en mayor o menor nivel de profundización, dependiendo de su complejidad e importancia, para que usted pueda actualizarse de la versión anterior a la actual. Para algunas features y por sus necesidades particulares, usted requerirá más información. En esos casos le recomendamos acceder al wiki y complementar lo visto.

Aquí le presentamos las diferentes fuentes de documentación a las que puede acceder.

GeneXus™

Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications