

Ejecutando la aplicación por primera vez

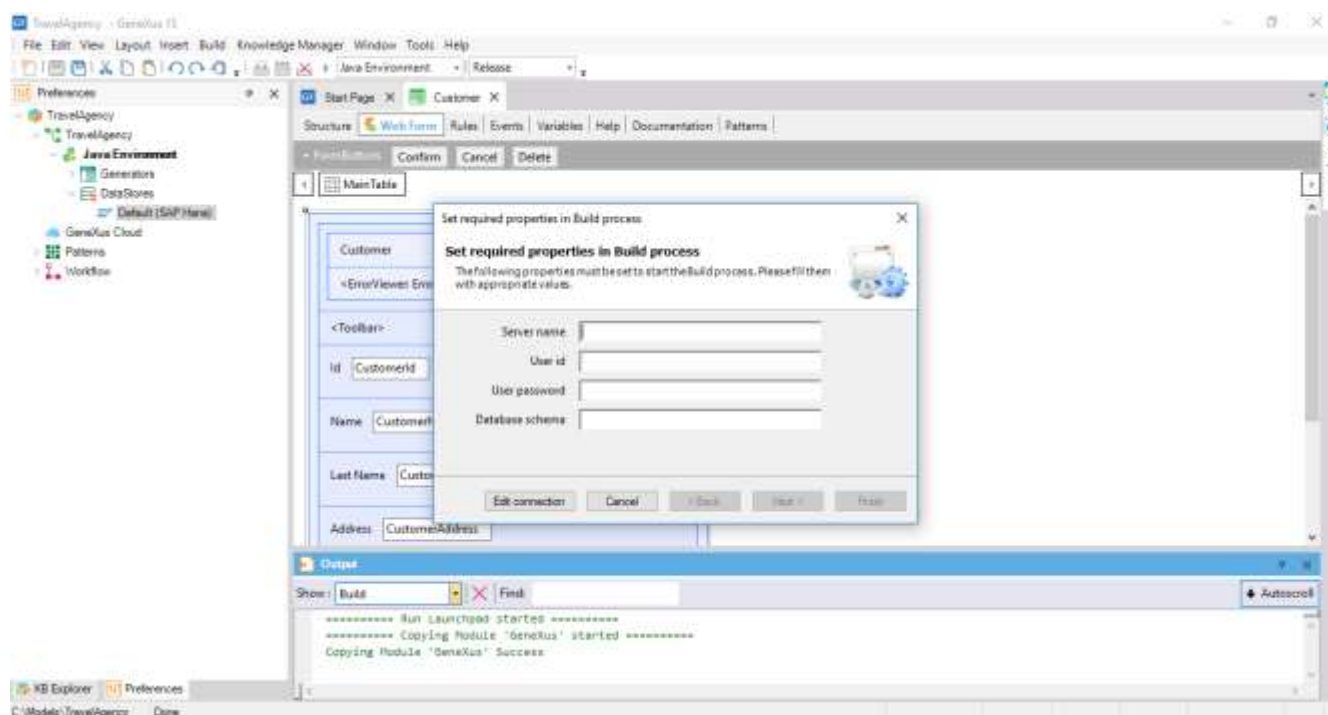
Queremos probar la transacción que creamos en los pasos anteriores. Para ello, le vamos a pedir a GeneXus que construya la aplicación, que cree la base de datos, que haga todo lo que tenga que hacer para tener los programas ejecutables para que podamos probar. A ese ciclo de pruebas en el que un desarrollador se embarca antes de tener la aplicación lista le llamamos **prototipación**.

En la versión full de GeneXus se le permitirá prototipar la aplicación tanto en **forma local** (es decir, los programas y la base de datos estarán en servidores de su red) como en **una nube** que GeneXus tiene a disposición. Además, podremos elegir hacerlo en cualquiera de los lenguajes de programación disponibles.

Pero en la versión especializada para SAP Systems, la base de datos será SAP HANA, el lenguaje Java, y la prototipación será en un server Tomcat que tengamos disponible. En nuestra demo será uno local.

Para configurar la información necesaria de la base de datos HANA y el web server, debemos ir a la solapa de Preferences, o directamente presionamos F5. Esta tecla de función será sumamente utilizada, pues con ella pedimos que se generen todos los objetos que se necesiten, que se compile el main y que se ejecute.

Por ser la primera vez que presionamos F5, se nos abre esta ventana donde se nos pide la información del server y base de datos:

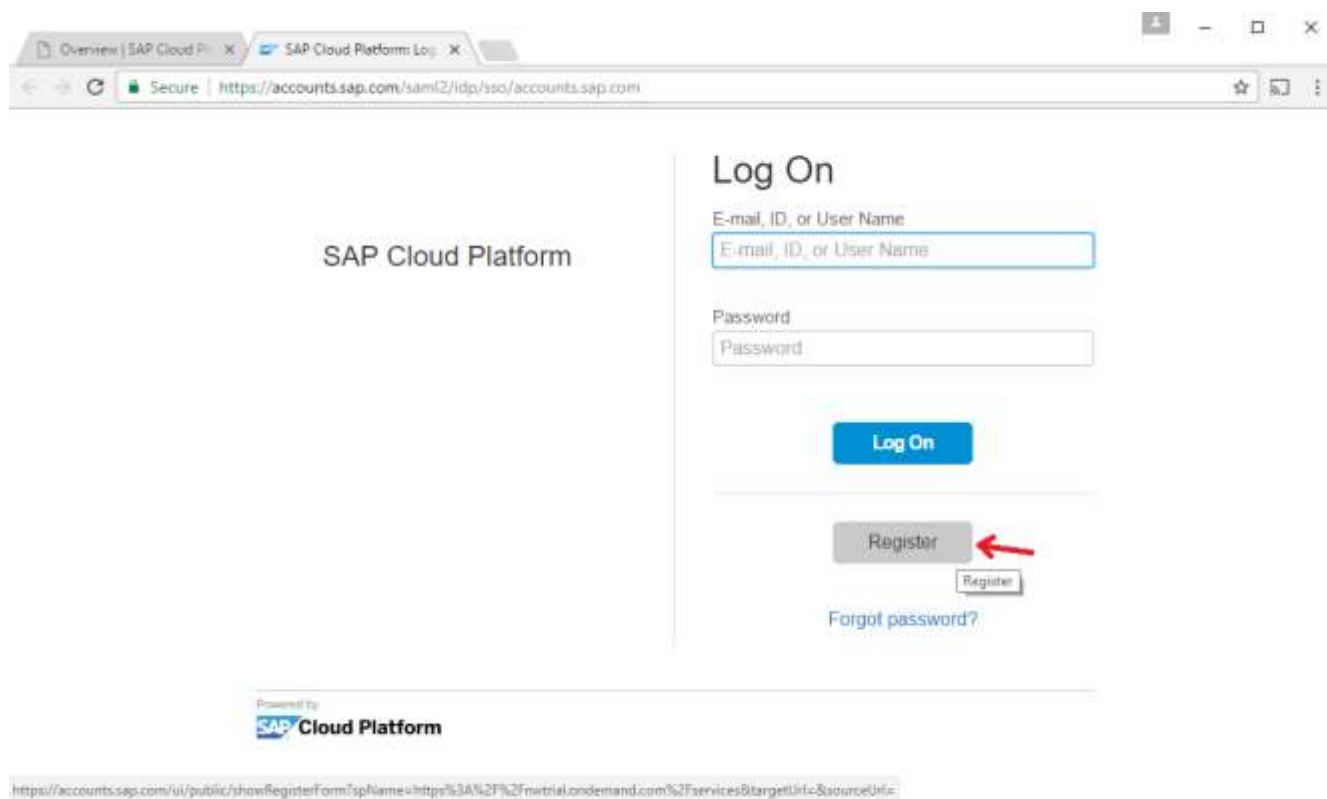


Una forma de construir aplicaciones que usen la base de datos SAP HANA es a través de la plataforma Cloud disponible para SAP, llamada SAP Cloud Platform.

Para propósitos de prueba, podemos crear una cuenta en hcp.sap.com.

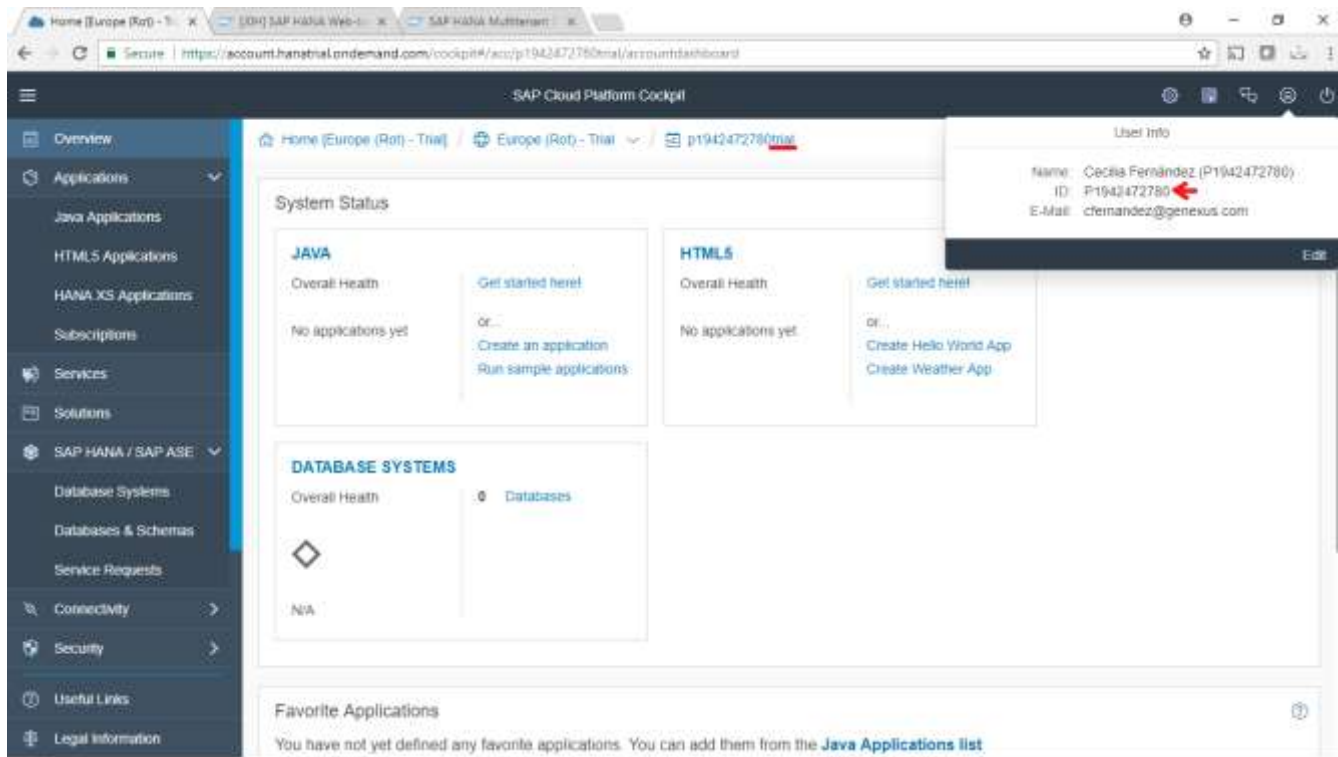


Aquí nos registramos con nuestra cuenta de email:



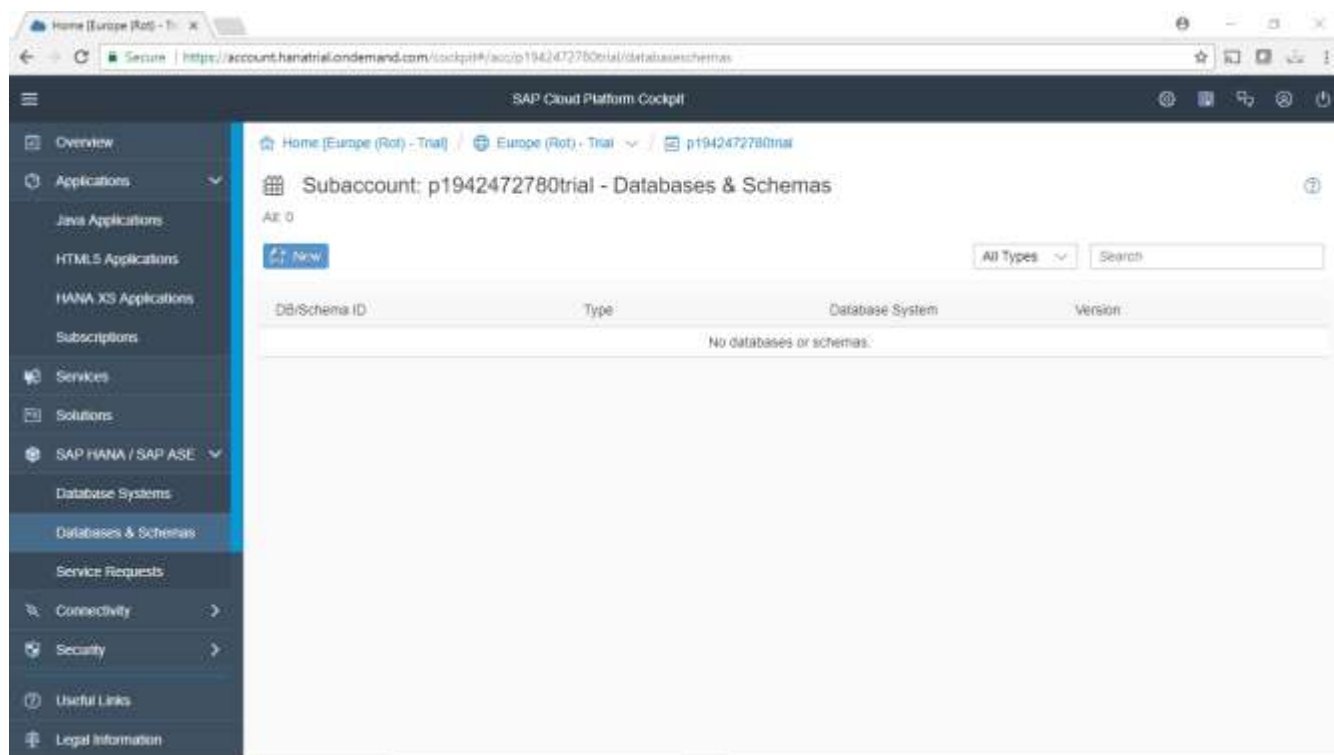
Si ya tenemos cuenta creada, nos logueamos con esa cuenta.

Elegimos la Neo Trial y podemos ver que el usuario se identifica con un ID, y que el nombre de la cuenta se compone de ese id de usuario, empezando en minúscula y la palabra trial.



La trial maneja un contenedor de bases de datos multitenant. Por cada cuenta se podrá crear una única base de datos dentro de ese contenedor.

Para hacerlo, vamos a la opción “Databases & Schemas”.



Creemos la base de datos para nuestra aplicación Travel Agency.

Debemos darle un nombre (Database ID). Por defecto se creará un usuario SYSTEM, que será el administrador y que será el que utilizaremos para prototipar nuestra aplicación. Debemos elegir una contraseña para ese usuario de la base de datos (debe ser de 15 caracteres o más) [SYSTEMUserPsw7!]:

Home [Europe (Rot) - Trial] / Europe (Rot) - Trial / p1942472780trial

Subaccount: p1942472780trial - Databases & Schemas

All: 0

[New](#)

DB/Schema ID

New Database/Schema

*Database ID:

Database System:

*SYSTEM User Password:

*Confirm Password:

Configure a User for SHINE.: ☐

Parameters

Web Access: ☒

DP Server: ☐

[Save](#) [Cancel](#)

Al grabar, empezará el proceso de creación de la base de datos, que demorará unos minutos.

Home [Europe (Rot) - Trial] / SAP HANA Multitenant / Cecilia Fernandez Costa

Secure | https://account.hanatrial.ondemand.com/vc/pickup/acc/p1942472780trial/travelagencydb/requests

SAP Cloud Platform Cockpit

Home [Europe (Rot) - Trial] / Europe (Rot) - Trial / p1942472780trial / travelagencydb

Database: travelagencydb - Events

All 17

Search

	Time	ID	Event	Description	User
✓	12 Mar 2018, 15:41:03	1	Create	Tenant DB creation finished successfully (created and configured)	P1942472780
✓	12 Mar 2018, 15:41:03	1	Change	HTTPS access from Internet enabled	P1942472780
✓	12 Mar 2018, 15:41:01	1	Change	Web dispatcher configuration added successfully	P1942472780
✓	12 Mar 2018, 15:41:01	1	Start	Database started successfully	P1942472780
✓	12 Mar 2018, 15:40:26	1	Start	Starting database	P1942472780
✓	12 Mar 2018, 15:40:25	1	Change	scriptserver service is disabled	P1942472780
✓	12 Mar 2018, 15:40:25	1	Change	dpserver service is disabled	P1942472780
✓	12 Mar 2018, 15:40:25	1	Change	Security settings configured successfully	P1942472780
✓	12 Mar 2018, 15:40:24	1	Change	Keystore updated successfully	P1942472780
✓	12 Mar 2018, 15:40:23	1	Stop	Database stopped successfully	P1942472780
✓	12 Mar 2018, 15:40:06	1	Stop	Stopping database	P1942472780
✓	12 Mar 2018, 15:40:06	1	Change	Statistics server checks are disabled	P1942472780
✓	12 Mar 2018, 15:40:05	1	Create	Database created successfully	P1942472780
✓	12 Mar 2018, 15:40:05	1	Create	Step 3 of 3 finished	P1942472780

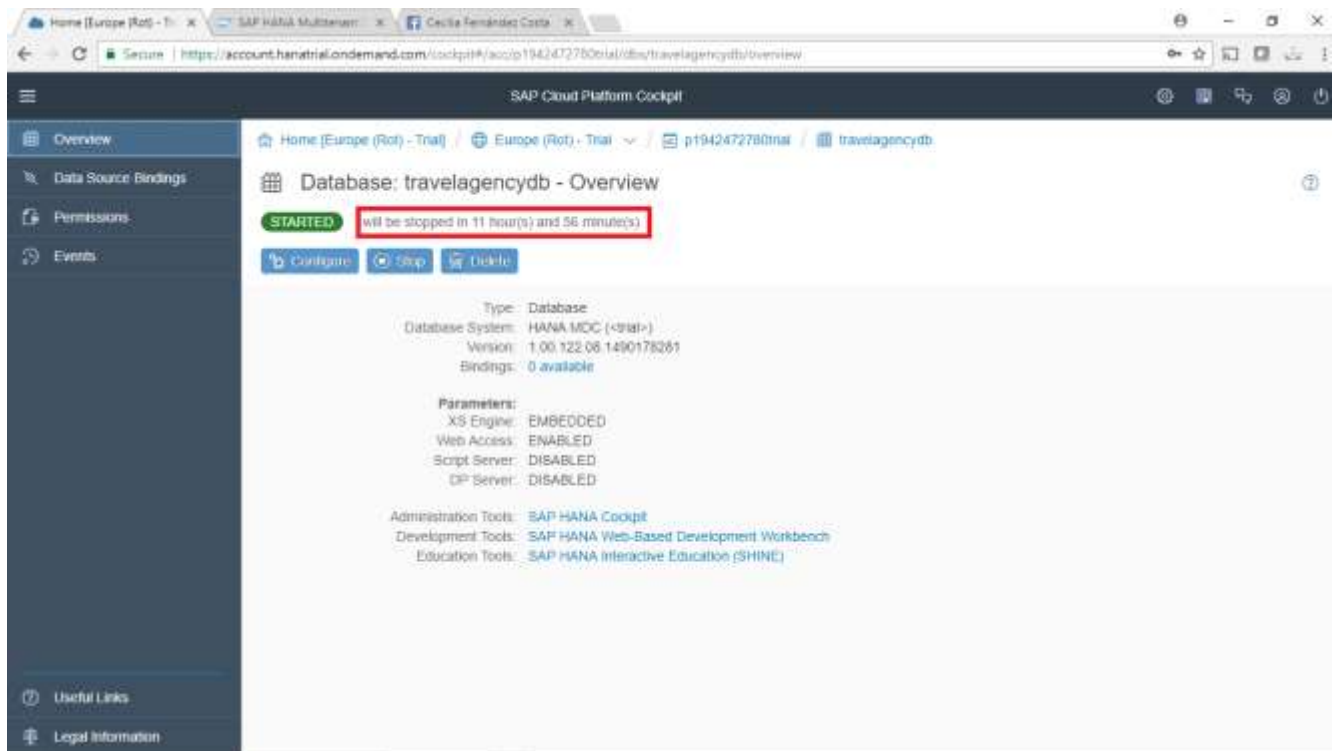
Useful Links

Legal Information

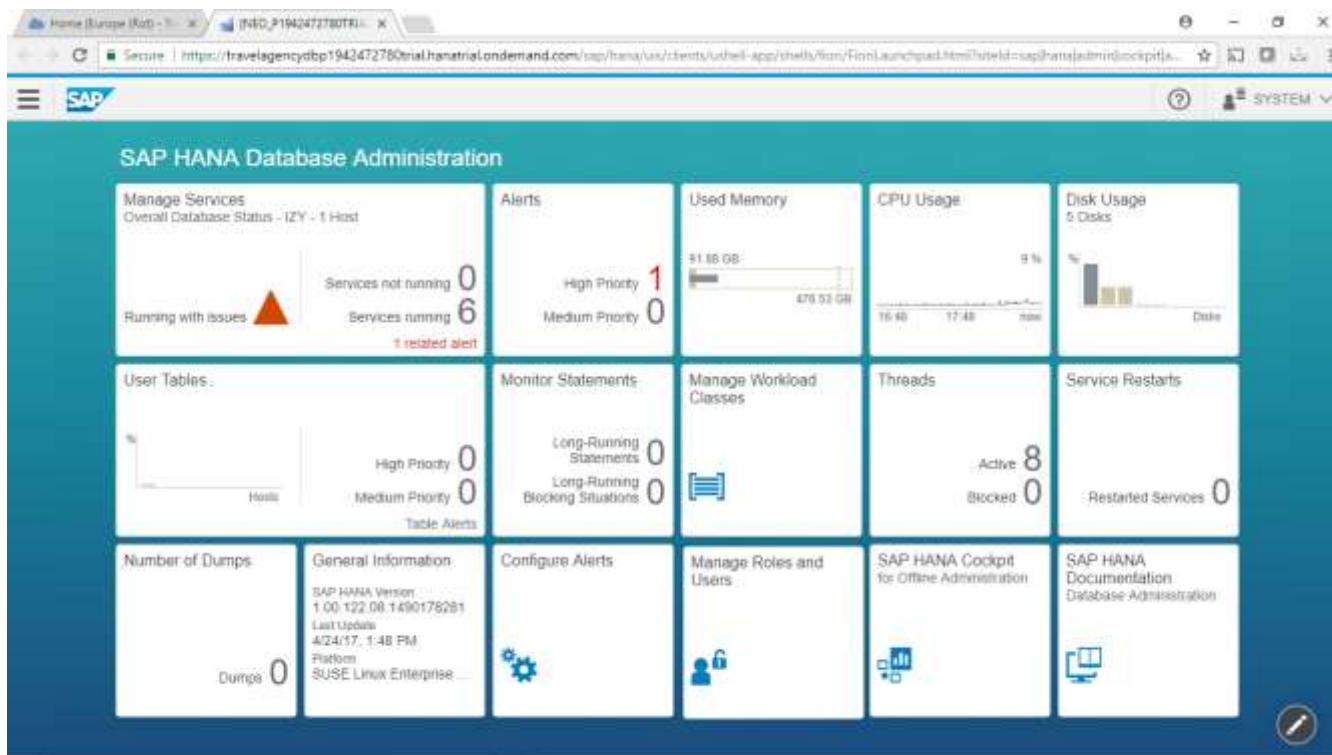
Al finalizar, yendo a Overview, veremos que la base de datos quedó levantada, ejecutándose, y que será

Video filmado con GeneXustm15 para SAP® systems – upgrade8

detenida en el tiempo que aquí se indica. Tendremos que levantarla pasado ese tiempo, si queremos utilizarla:

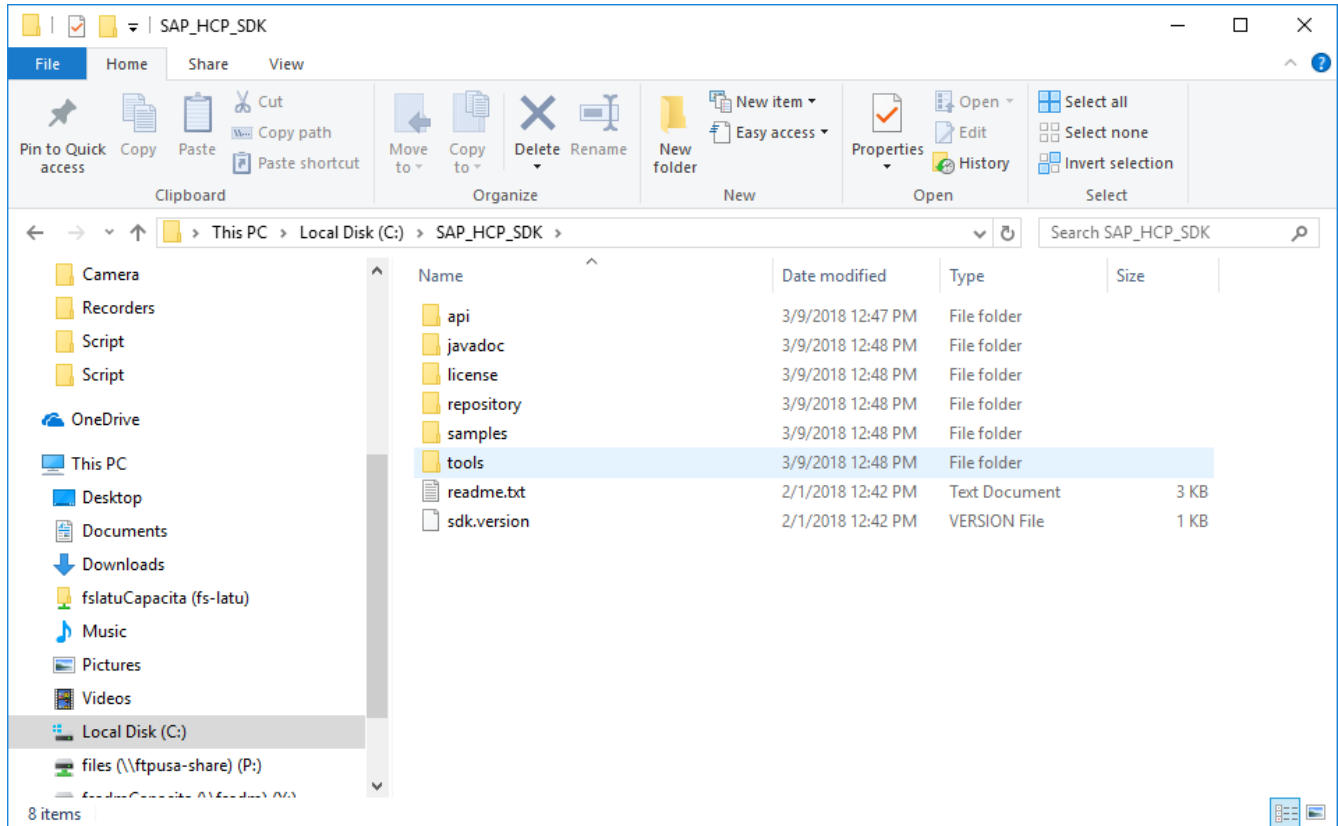


Yendo a SAP HANA Cockpit se creará el usuario SYSTEM administrador, y aparecerán estas herramientas de administración, entre ellas la posibilidad de crear nuevos usuarios y roles.



Ahora bien, para poder conectarnos a través de JDBC a esta base de datos que acabamos de crear debemos abrir un “túnel”.

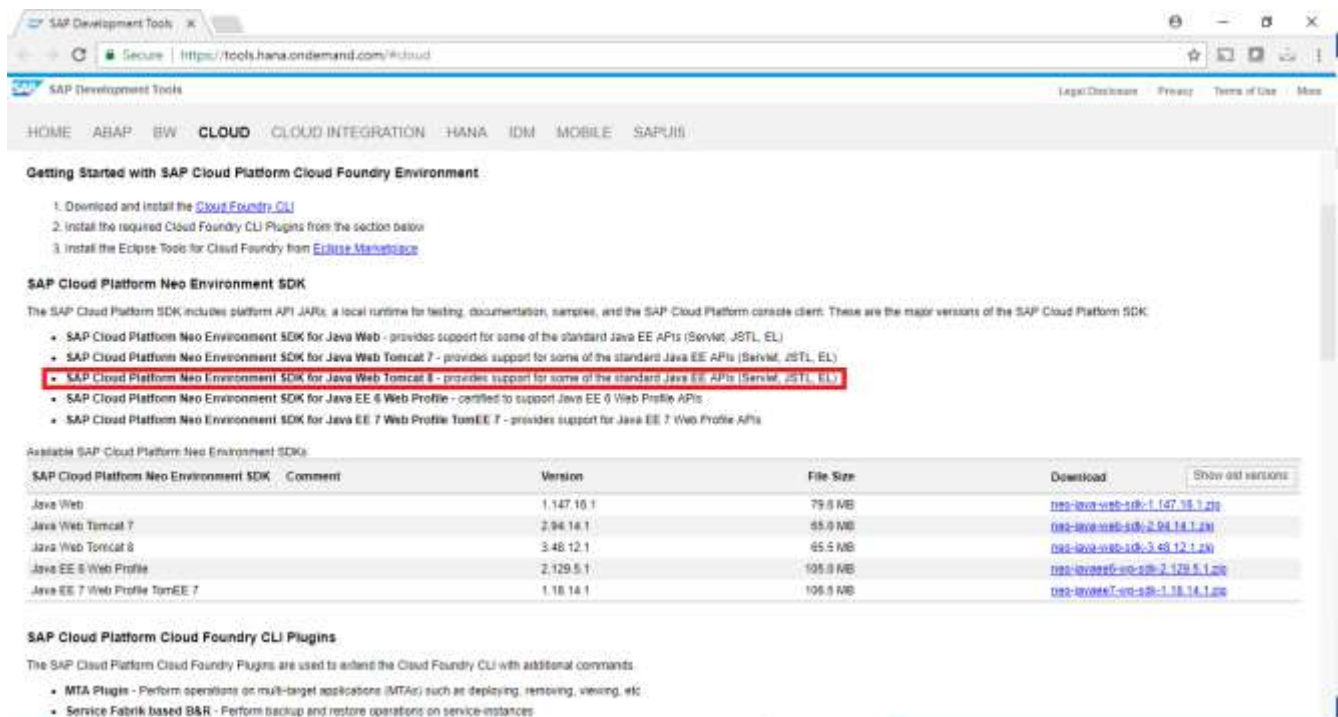
Para ello, teniendo instalado el SDK del SAP Cloud Platform que corresponda a la versión del Java Web Tomcat que estemos utilizando:



(Se descarga desde aquí:

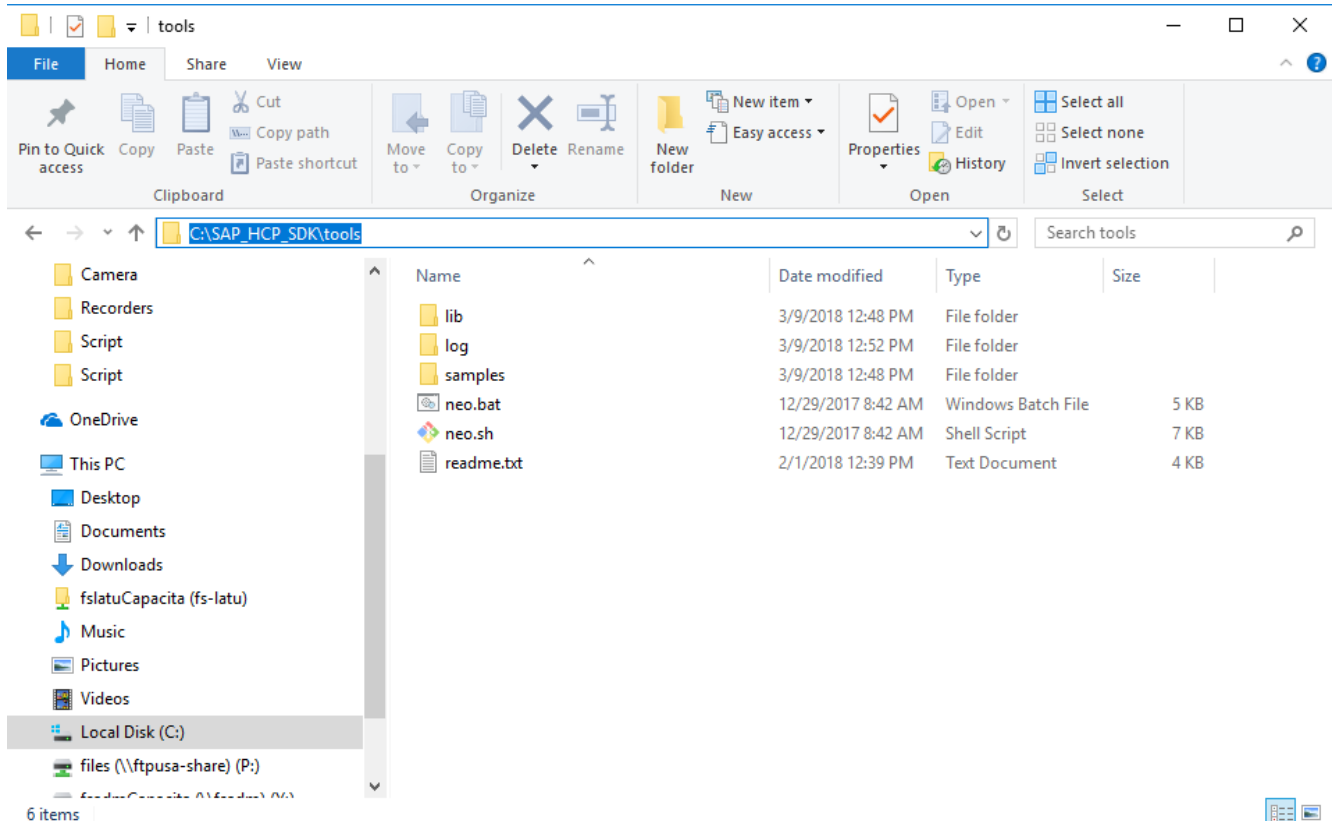
<https://tools.hana.ondemand.com/#cloud>

Nosotros tenemos el Tomcat 8, por lo que nos descargamos este de aquí:



Allí se contará con todo lo necesario para construir aplicaciones en SAP Cloud Platform, como bibliotecas requeridas para la compilación de aplicaciones SAP Cloud Platform, documentación, ejemplos, tools para uso por línea de comandos).

Debemos abrir una consola Command Prompt sobre el folder tools:

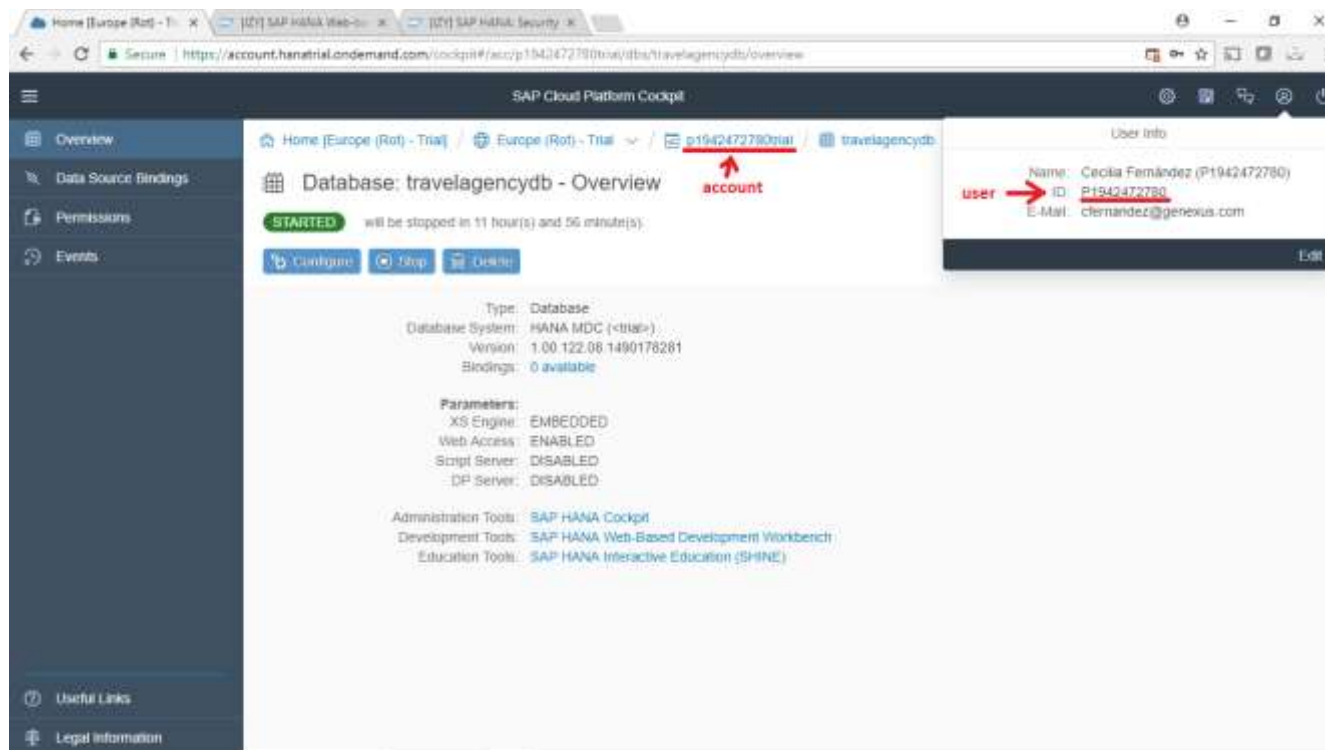


Y allí utilizar el comando:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\SAP_HCP_SDK\tools>neo open-db-tunnel -h <host> -u <user> -a <subaccount> -p <password> -i <schema id>
```

Donde “user” y “account” son los que vimos antes, correspondientes a nuestro usuario en el Cockpit:



Por lo que sustituimos en el comando esos valores...

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\SAP_HCP_SDK\tools>neo open-db-tunnel -h <host> -u P1942472780 -a p1942472780trial -p <password> -i <schema id>
```

El host en nuestro caso en el que estamos utilizando la trial es: hanatrial.ondemand.com

El password es el de nuestra cuenta, con el que nos logueamos en el sitio web.

Y el schema id es el de la base de datos que acabamos de crear: travelagencydb

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\SAP_HCP_SDK\tools>neo open-db-tunnel -h hanatrial.ondemand.com -u P1942472780 -a p1942472780trial -p Segundo7! -i travelagencydb
```

Observemos que como resultado de la ejecución del comando se ha abierto el tunnel, que permanecerá abierto por 24 hrs, a menos que lo cerremos explícitamente antes.



```
SAP Cloud Platform Console Client
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\SAP_HCP_SDK\tools>neo open-db-tunnel -h hanatrial.ondemand.com -u P1942472780 -a p1942472780trial -p [REDACTED] -i travelagencydb

SAP Cloud Platform Console Client

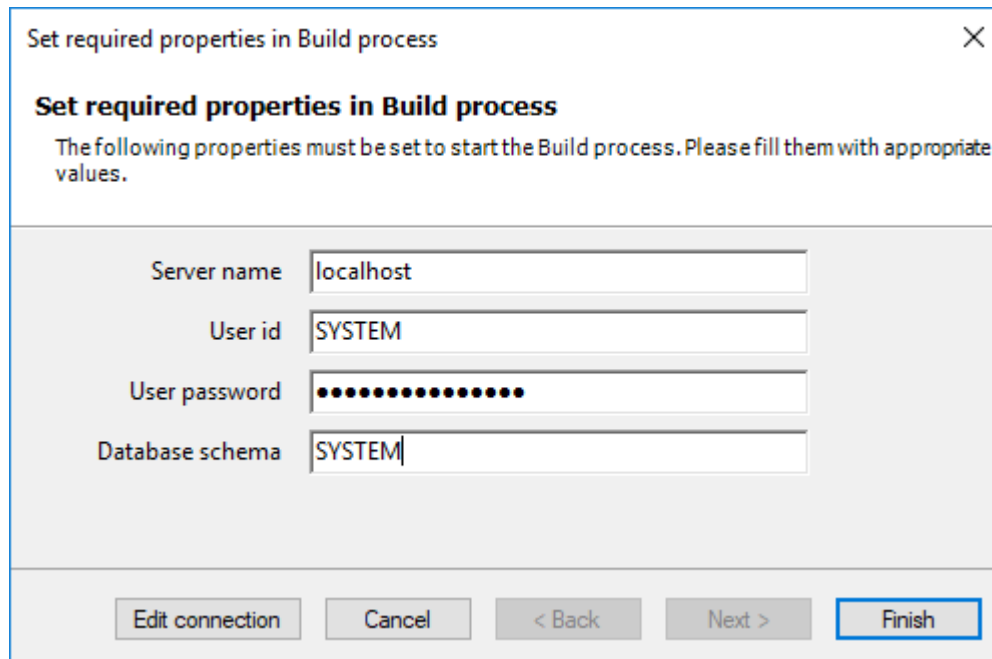
Opening tunnel...
Tunnel opened.

Use these properties to connect to your schema:
Host name      : localhost
Database type  : HANAMDC
JDBC Url       : jdbc:sap://localhost:30015/
Instance number : 00

Use any valid database user for the tunnel.
This tunnel will close automatically in 24 hours on Mar 13, 2018 at 4:18 PM or when you close the shell.
Press ENTER to close the tunnel now.
```

Ahora sí, volviendo a GeneXus...

El nombre del servidor será localhost, pues es el valor de Host Name que nos da al abrir el túnel.



Set required properties in Build process

The following properties must be set to start the Build process. Please fill them with appropriate values.

Server name	localhost
User id	SYSTEM
User password	
Database schema	SYSTEM

El user será SYSTEM

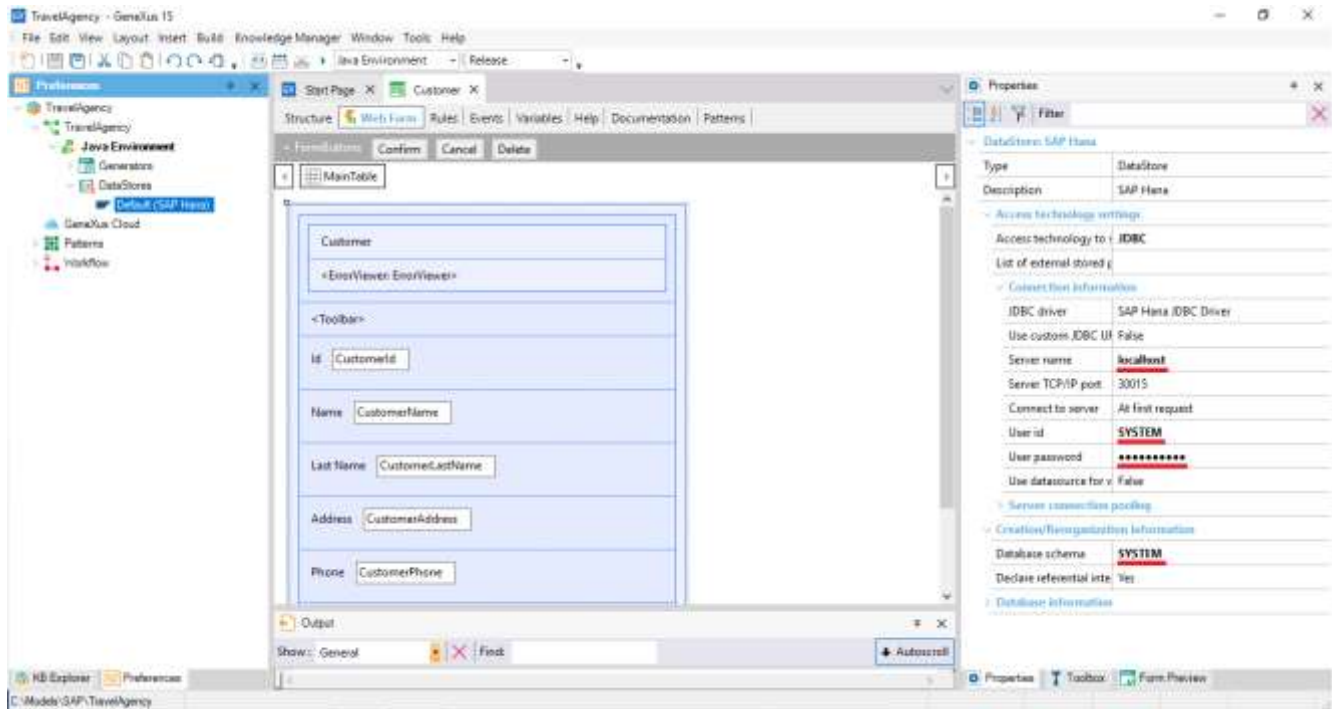
La Password será la que indicamos para ese usuario al crear la base de datos en la trial de SAP Cloud Platform hace unos instantes...

Y el esquema de base de datos se identifica con el mismo nombre que el usuario, SYSTEM.

Si ahora presionamos Finish, GeneXus construirá la aplicación a partir de un menú propio del desarrollador,

para ir probando todos los objetos que se van creando, sin preocuparnos de programar la invocación.

Presionemos Cancel. Yendo a la solapa Preferences, veremos que de todos modos la información que acabamos de ingresar ya quedó configurada en las propiedades correspondientes. Así, si vamos a la información de los DataStores, vemos que tenemos el Default (SAP Hana) y si activamos la ventana de propiedades (con F4, o aquí a la derecha):



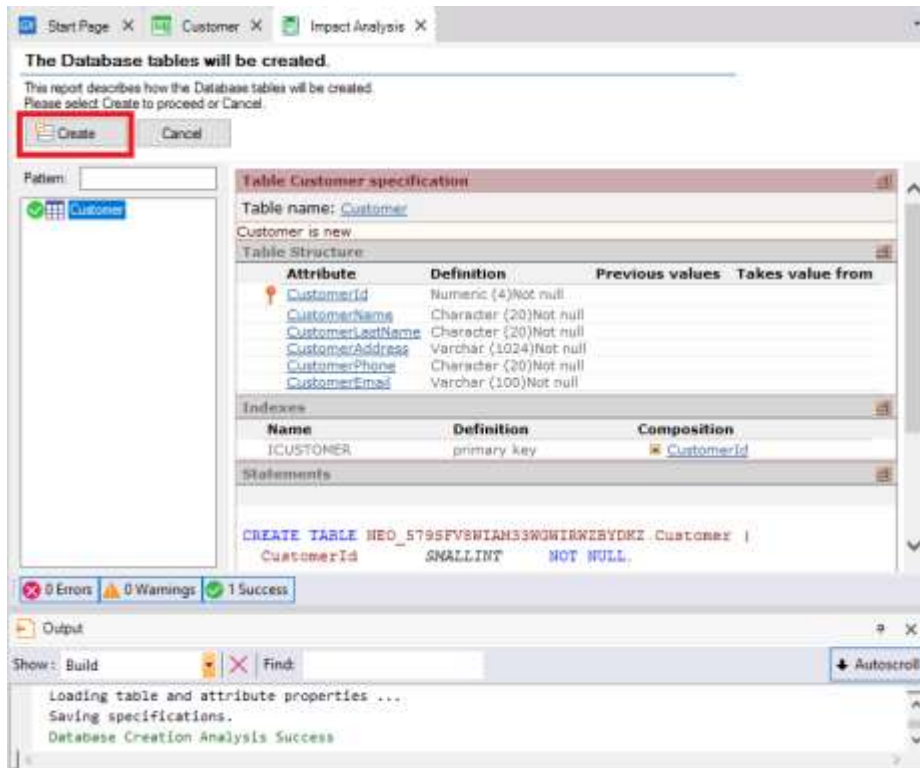
¡aquí están!

Lo único que nos resta hacer para poder ejecutar ese menú del desarrollador del que hablábamos es presionar F5.

Vemos que la ventana de Output informa que se inició la ejecución del Developer Menú, que finaliza con un informe que recibe el nombre "Impact Analysis", Análisis de Impacto.

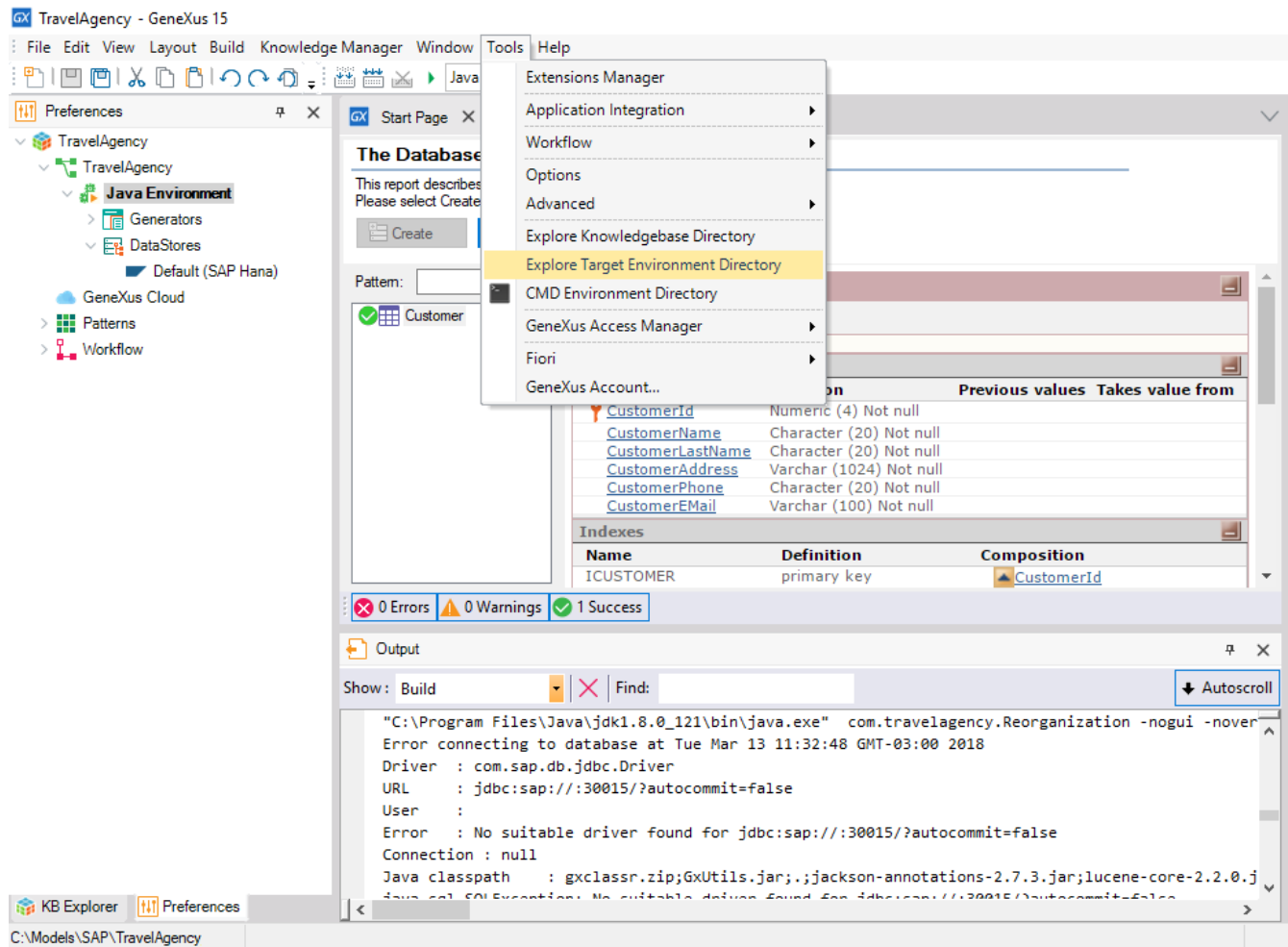
más adelante explicaremos este concepto.

Dado que estamos de acuerdo con que se cree esta tabla con esta definición en la base de datos, presionamos el botón CREATE....



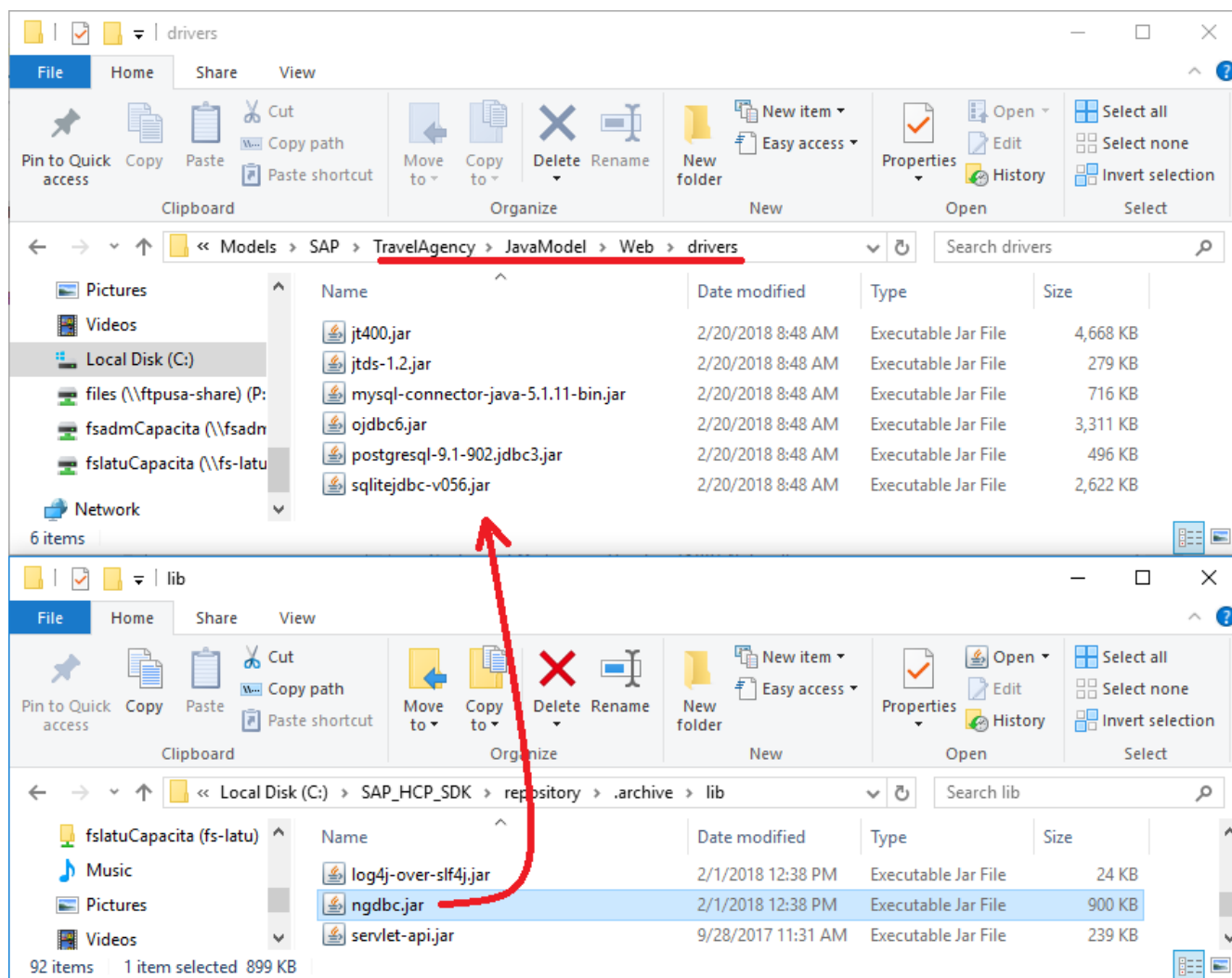
y GeneXus procede a crear los programas necesarios para crear la tabla **Customer** en la base de datos con esta estructura.

A continuación GeneXus ejecuta estos programas...

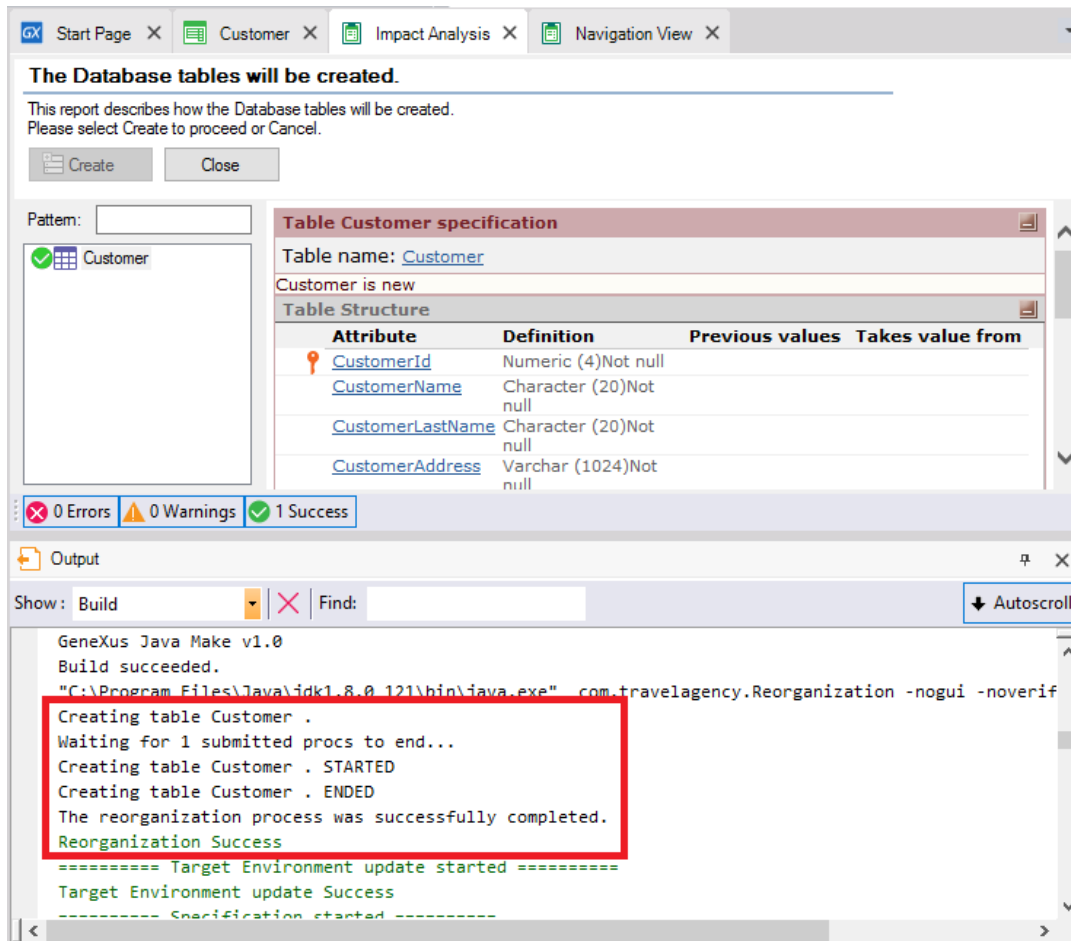


El driver se llama ngdbc.jar ¿De dónde lo sacamos? Ese archivo usualmente viene con el SAP HANA Client. Aquí la guía de instalación: <https://help.sap.com/viewer/e7e79e15f5284474b965872bf0fa3d63/2.0.02/en-US>

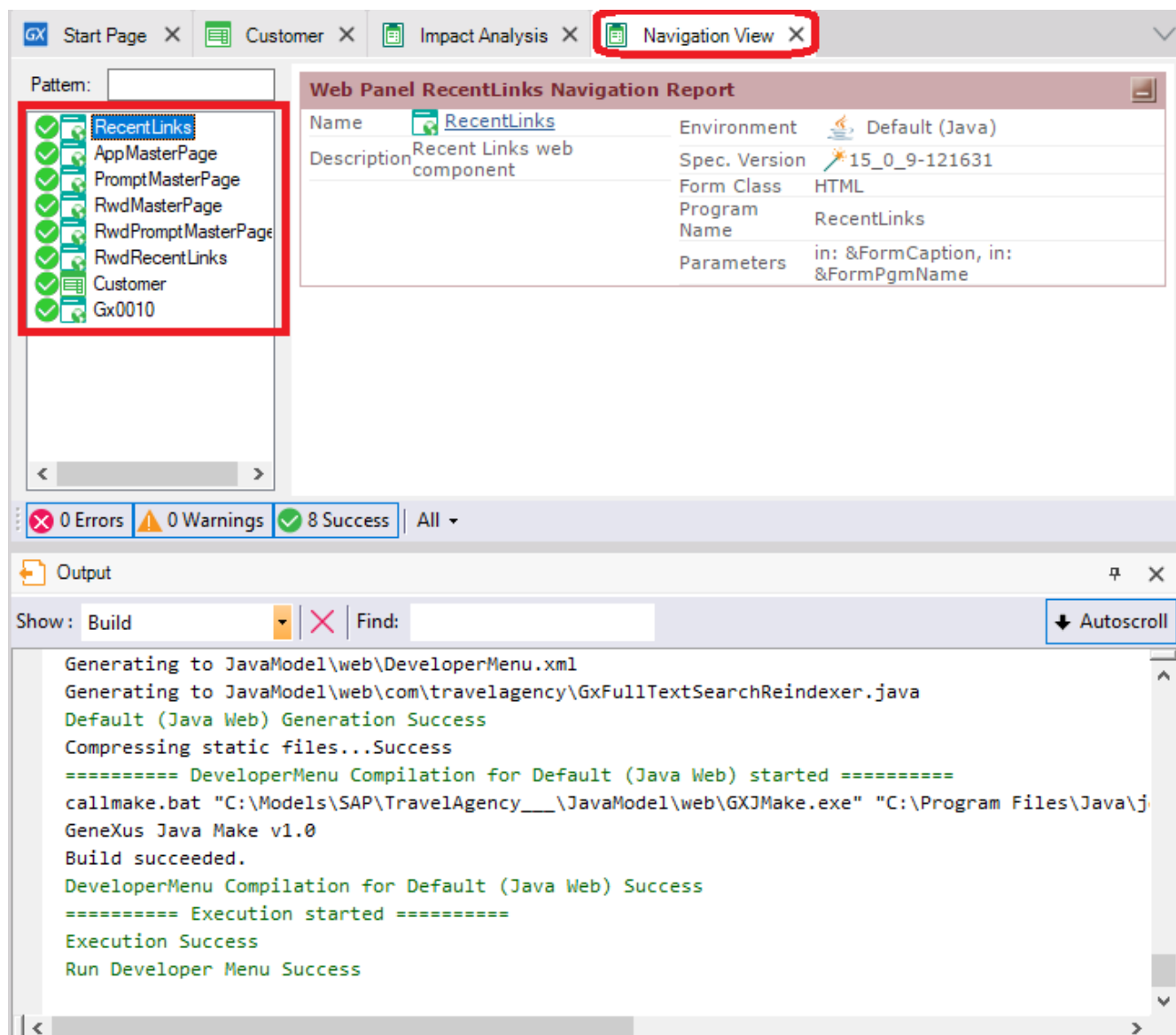
Nosotros lo tenemos aquí:



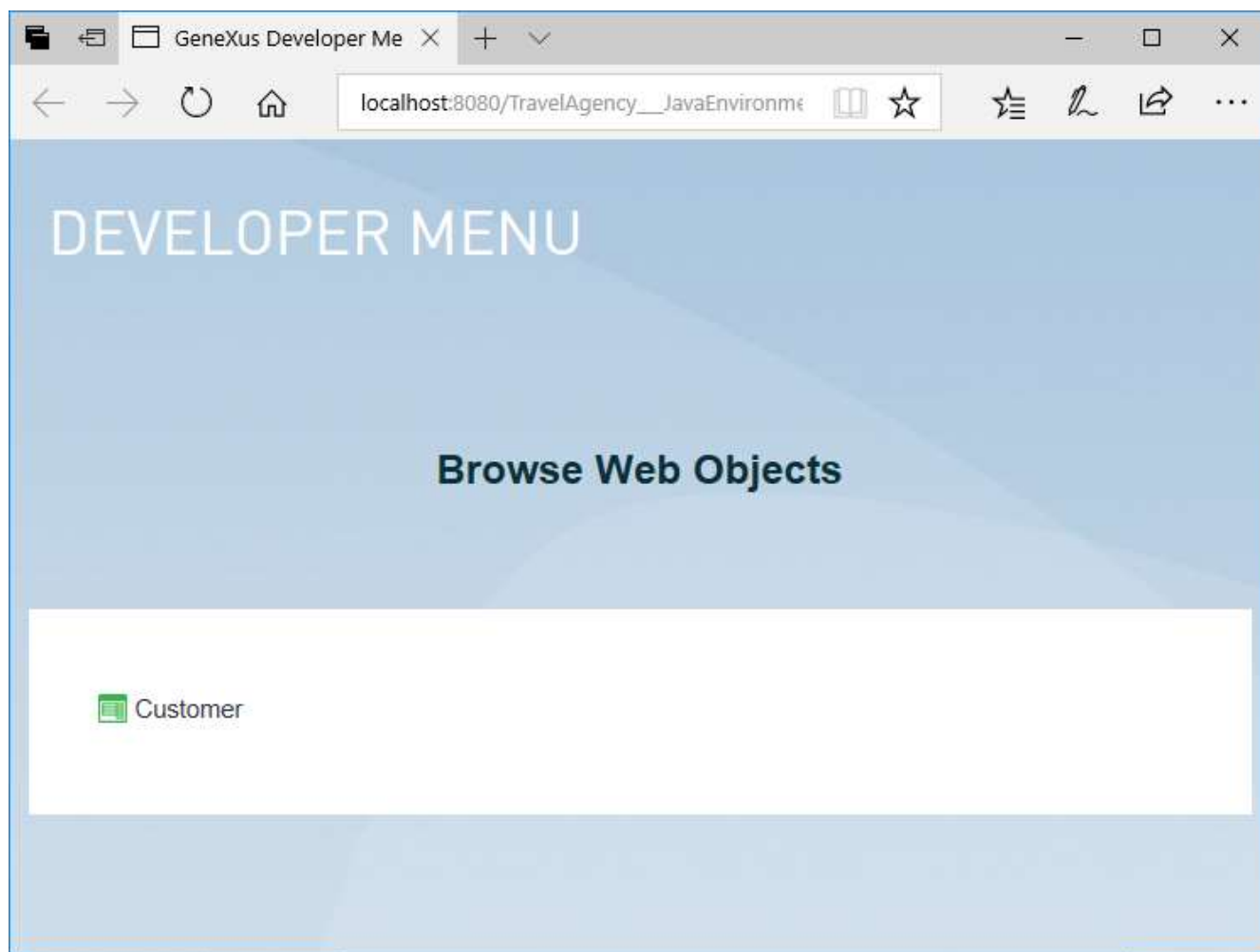
Ahora sí, volvemos a presionar F5, y Create.



y queda creada la tabla Customer en la base de datos. GeneXus continúa generando otros programas (es decir todas las líneas de código necesarias en el lenguaje de programación elegido para que nuestra aplicación haga lo que deseamos, en este caso ingresar datos de clientes).... y nos muestra si el resultado fue exitoso, o surgió algún error o advertencia.



Aquí tenemos la aplicación en funcionamiento:



Vemos que tenemos la transacción **Customer** que hemos definido.

Pulsemos el botón derecho del mouse sobre el link y seleccionemos ejecutarlo en una nueva pestaña.

Aquí estamos viendo una página que nos permite ingresar, modificar y eliminar clientes. Ingrese los datos de nuestro primer cliente....

GeneXus Developer Men x Customer x

localhost:8080/TravelAgency_JavaEnvironment/servlet/com.travelagency.customer

Application Name

Customer

Navigation: << < > >> SELECT

Id:

Name:

Last Name:

Address:

Phone:

Email:

Buttons: Confirm Cancel Delete

Este cliente se identificará con el número 1

Se llamará John:

Su apellido es Smith:

Vive en la quinta avenida

Su teléfono es: 1111

Y su correo electrónico es jsmith@hotmail.com

GeneXus Developer Men x Customer x

localhost:8080/TravelAgency_JavaEnvironment/servlet/com.travelagency.customer#

Application Name

Customer

Id 1

Name Jhon

Last Name Smith

Address 5th. Avenue 1111

Phone 1111

Email jsmith@hotmail.com

Confirm Cancel Delete

Ahora pulsamos el botón Confirm:

Y un mensaje nos avisa que los datos han sido agregados satisfactoriamente, mientras que el formulario se limpia y queda listo para un nuevo ingreso:

GeneXus Developer Men x Customer x

localhost:8080/TravelAgency_JavaEnvironment/servlet/com.travelagency.customer#

Application Name

Customer

• Data has been successfully added.

Id

Name

Last Name

Address

Confirm Cancel Delete

Ingresemos un 2do cliente... lo identificamos con el número 2:

Se llama Susan:

Su apellido es "Brown":

Su dirección es en la séptima avenida

Su teléfono es 2222

Con correo electrónico sbrown@gmail.com

The screenshot shows a web browser window with two tabs: 'GeneXus Developer Men...' and 'Customer'. The address bar shows 'localhost:8080/TravelAgency_JavaEnvironment/servlet/com.travelagency.customer#'. The page title is 'Application Name'. The main content area is titled 'Customer' and contains a form with the following fields:

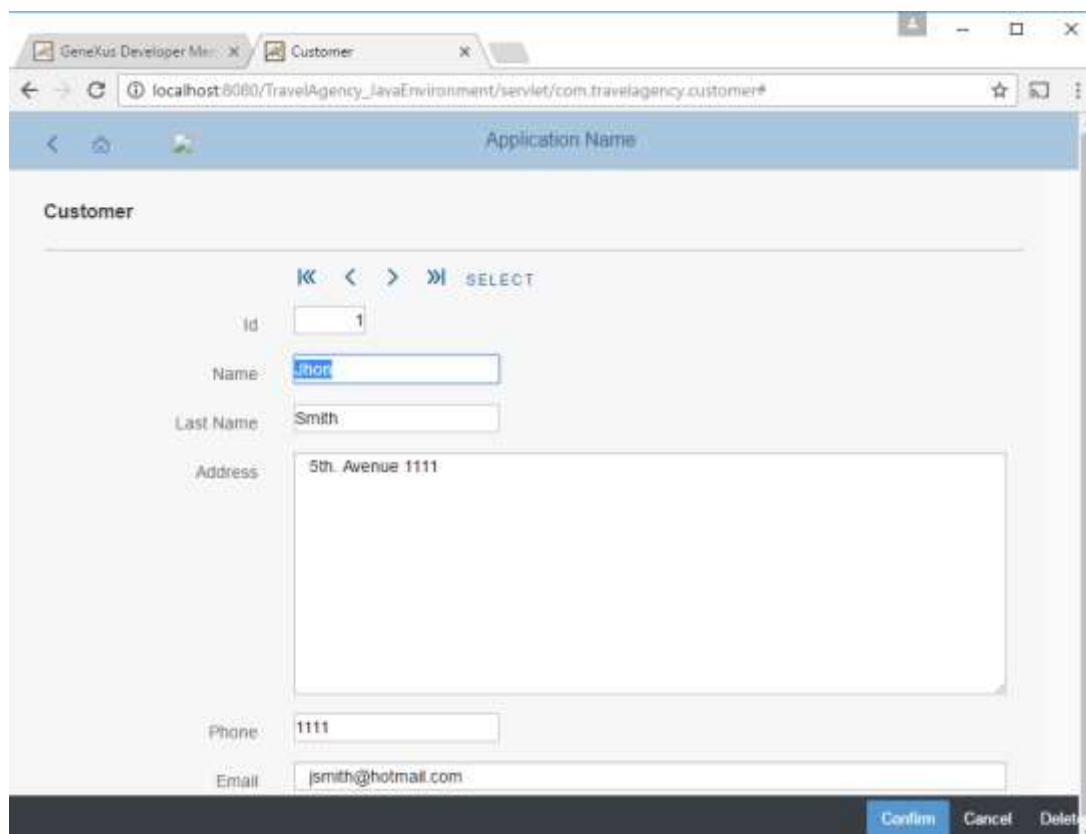
- Id: 2
- Name: Susan
- Last Name: Brown
- Address: 7th. Avenue 2222
- Phone: 2222
- Email: sbrown@gmail.com

At the bottom right of the form, there are three buttons: 'Confirm' (blue), 'Cancel' (gray), and 'Delete' (red).

Seleccionamos el botón Confirm:

Y nuevamente sale el mensaje que avisa que los datos han sido agregados satisfactoriamente.

Vamos a buscar los datos de John, digitamos su número de cliente (el uno) y luego presionamos tabulador.



Vemos que se muestran todos los datos de John y una vez que estamos posicionados aquí podríamos cambiar algún dato, por ejemplo su teléfono.

O podríamos eventualmente eliminar el cliente John del registro de clientes con el botón Delete.

También podemos usar los botones de aquí arriba, para navegar entre clientes, yendo por ejemplo al siguiente, al anterior, al primero o al último:

GeneXus Developer Men... Customer

localhost:8080/TravelAgency_JavaEnvironment/servlet/com.travelagency.customer#

Application Name

Customer

« < > » SELECT

Id 1

Name Jhon

Last Name Smith

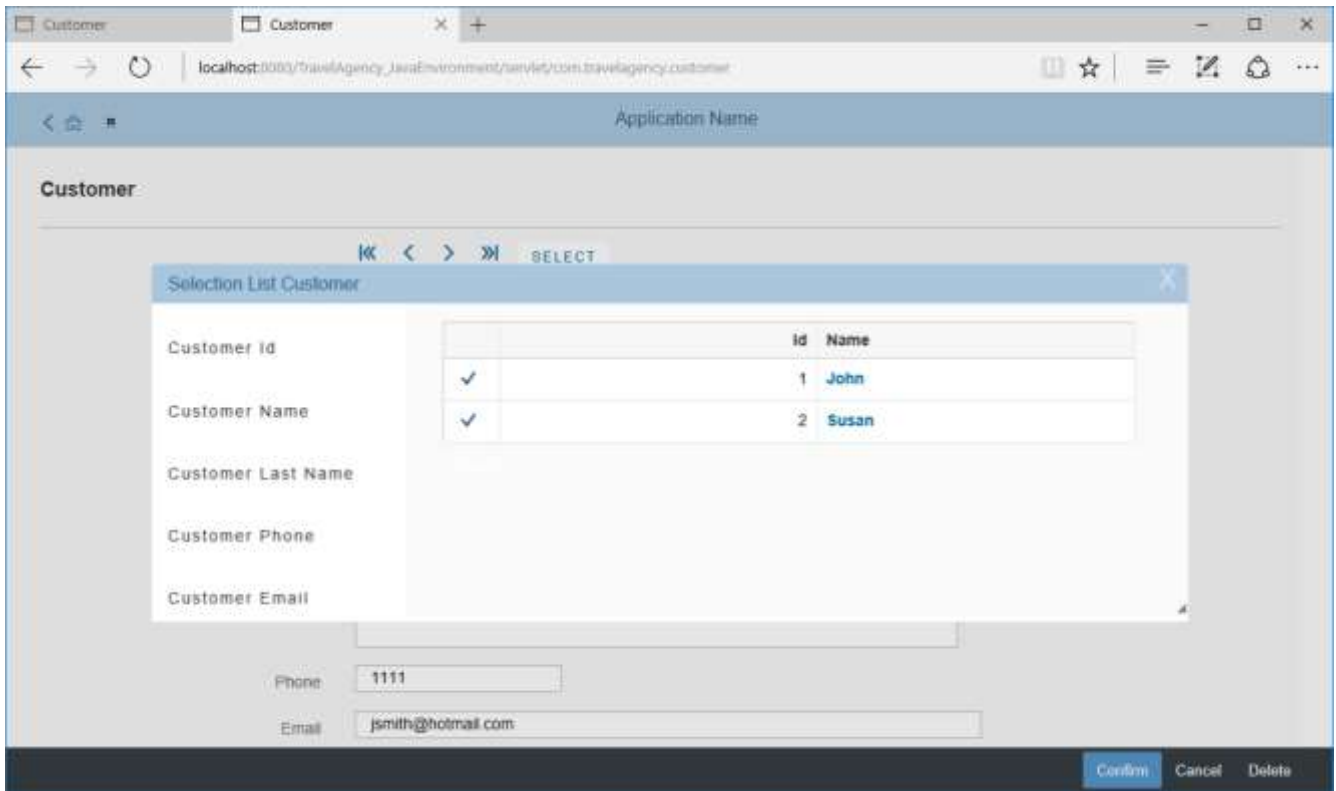
Address 5th. Avenue 1111

Phone 111122

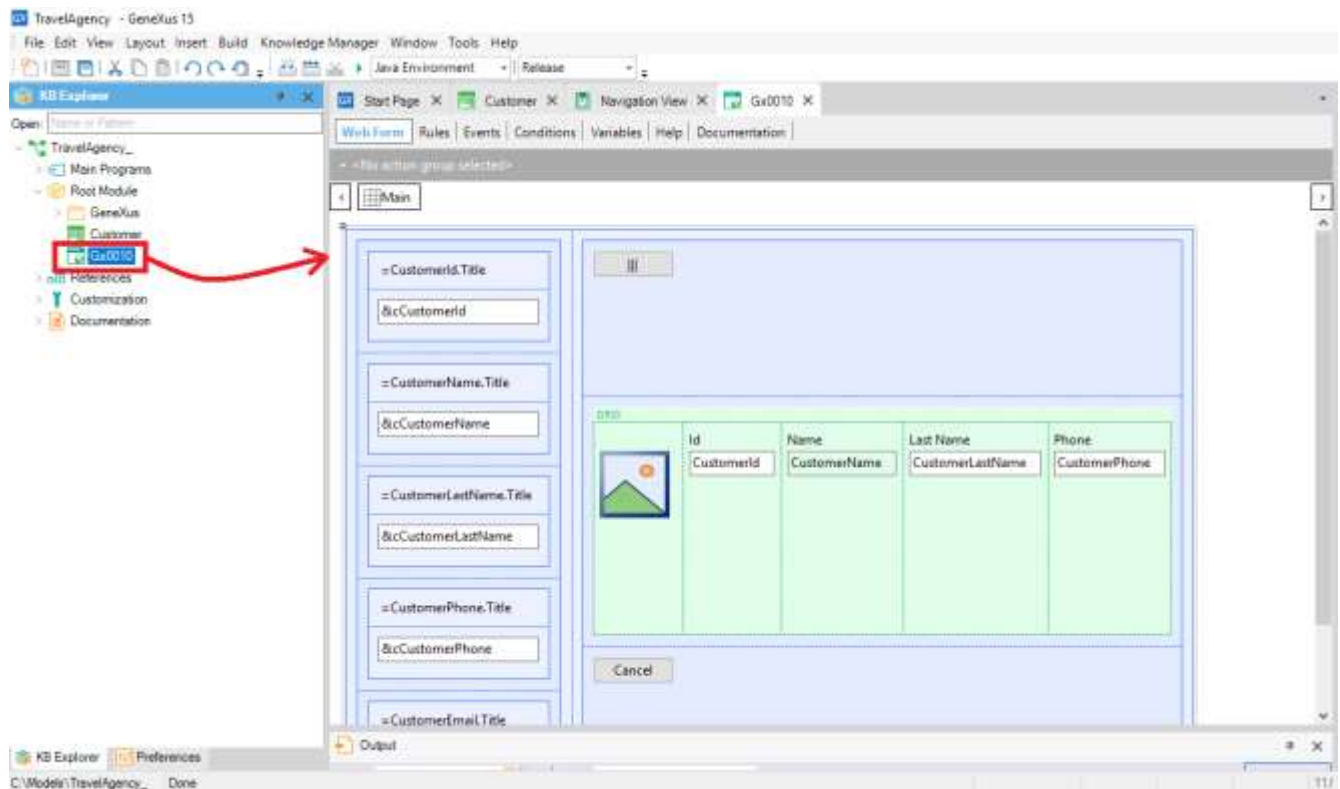
Email jsmith@hotmail.com

Confirm Cancel Delete

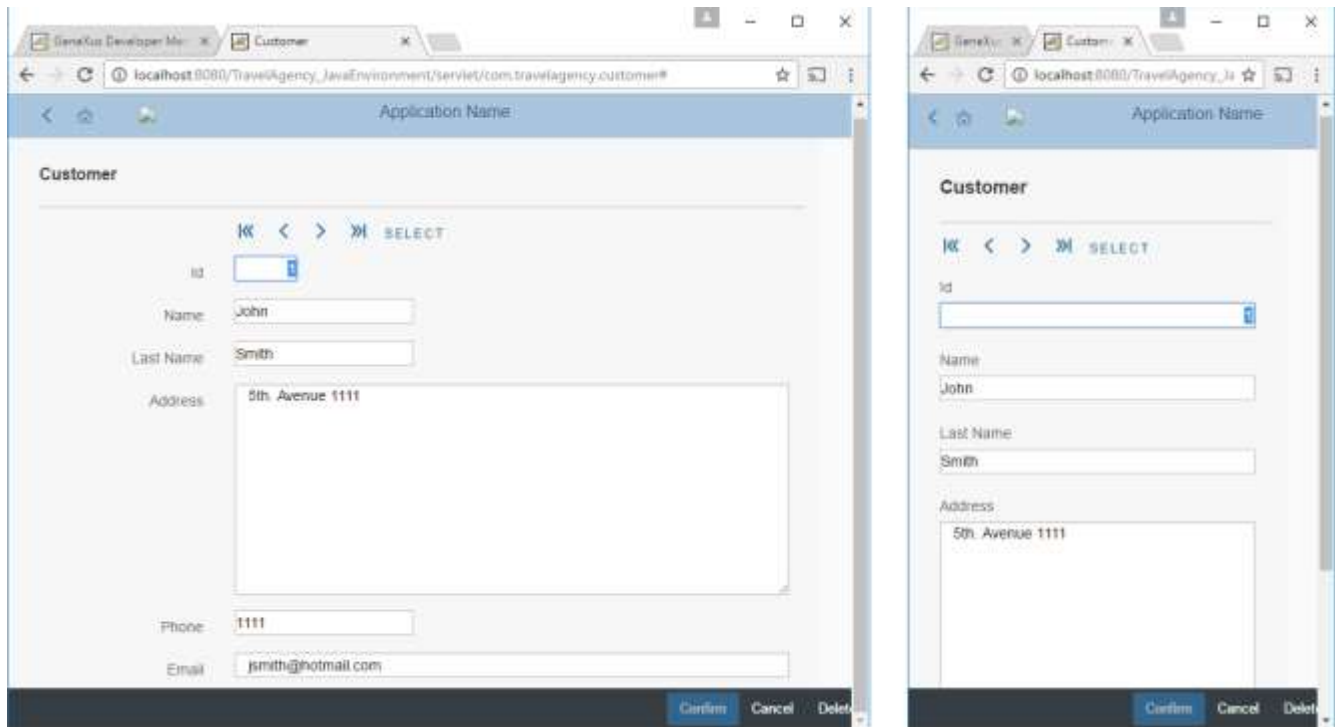
o para buscar a un cliente en particular mediante esta ventana de aquí:



Esa ventana fue creada automáticamente por GeneXus. Corresponde a este objeto:



Observemos que si ajustamos el tamaño de la pantalla, los controles se ajustan al espacio disponible.



¿Qué significa esto? GeneXus genera aplicaciones web responsive, o aplicaciones sensibles, que ofrecen una óptima visualización, buena navegación, y un excelente aprovechamiento de los espacios de la pantalla.

Todo esto se logra utilizando un diseño que permite ver toda la información de una forma armoniosa dentro de la pantalla, y GeneXus nos provee de funciones de edición para lograrlo. Las mismas nos brindan la posibilidad de mostrar diferentes formas de visualizar el layout dependiendo del tamaño de la pantalla del dispositivo.

Ahora que hemos visto en ejecución **todo lo que GeneXus generó automáticamente a partir de nuestra definición de la transacción Customer...** volvamos al ambiente de desarrollo.