



## Data Management

KeepCoding

**Fechas: 12 y 13 de diciembre de 2019.**

**Horas: 8h**

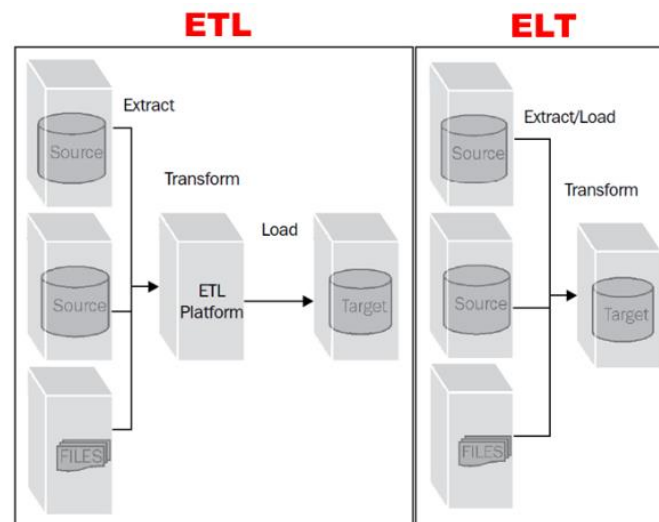
**Instructora: Nerea Sevilla**

**Nombre Material: Ejercicios\_ODI12c**

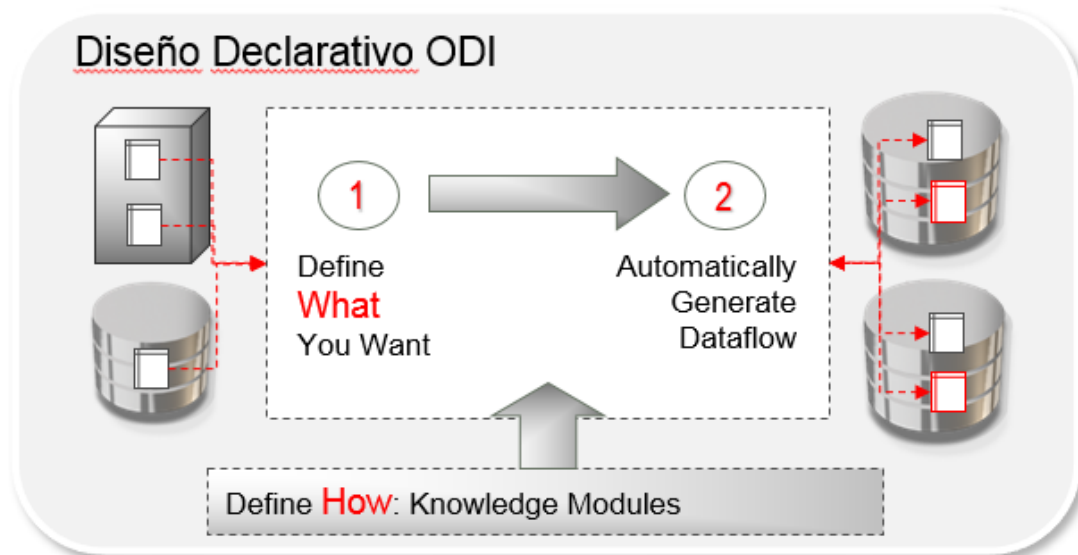
## Oracle Data Integrator 12c

La complejidad de fuentes de datos procedentes de diferentes sistemas, hace imprescindible contar las herramientas de integración de datos como Oracle Data Integrator (ODI) 12c. ODI 12c es una solución completa que proporciona integridad de datos ya que asegura la consistencia y exactitud de los datos para entornos operacionales y analíticos.

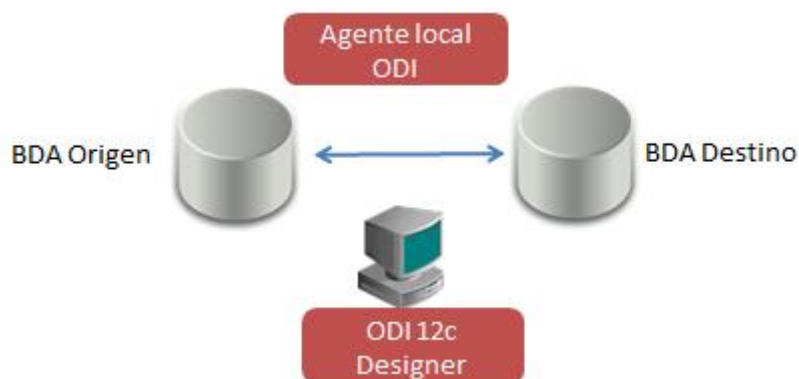
ODI12c emplea arquitectura EL-T, primero extrae los datos de las fuentes; luego carga las tablas en el servidor destino; y por último ocurre la transformación de los datos en destino utilizando sentencias SQL nativas.



Además, utiliza diseño declarativo, cuyas ventajas son que es un diseño simple e intuitiva por lo que reduce significativamente la curva de aprendizaje y acorta los plazos de implementación y mantenimiento de los procesos.



La parte práctica de ODI12c consistirá en el diseño de la arquitectura que nos permita realizar un movimiento de datos entre un origen y un destino.



Para simplificar y entender los conceptos que intervienen en las herramientas de integración de datos, moveremos datos entre bases de datos. Pero estos conceptos pueden ser extrapolados a movimientos entre diferentes fuentes origen y destino de datos bases de datos, del tipo que sean estructuradas como Oracle o no estructuradas como MongoDB, entre ficheros planos csv o xml, etc.... ya que ODI12c permite utilizar casi cualquier tipo de drivers dentro de las tecnologías que es capaz de manejar.

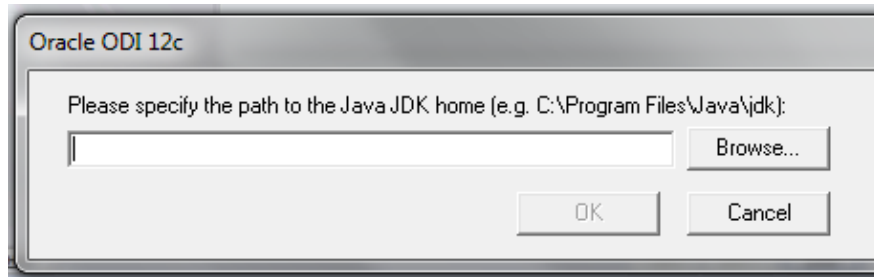
El objetivo de estos ejercicios en relación con los procesos de integración es:

- Aprender a definir arquitecturas físicas y lógicas
- Aprender a definir modelos de datos
- Aprender a diseñar procesos de integración de datos

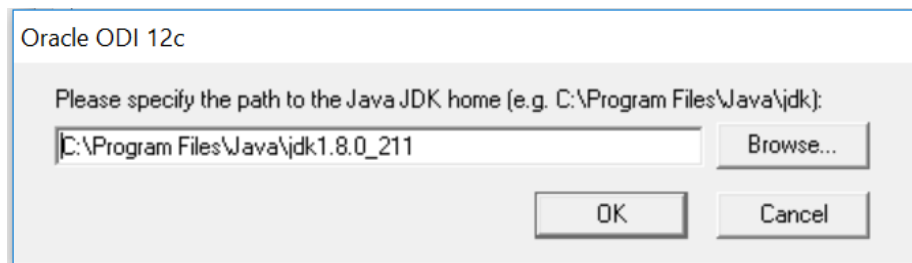
## Ejercicio 1.-Configurar acceso a Repositorios

Antes de conectar al repositorio de trabajo, debemos configurar el acceso al repositorio de trabajo creado mediante RCU utilizando ODI Studio 12c.

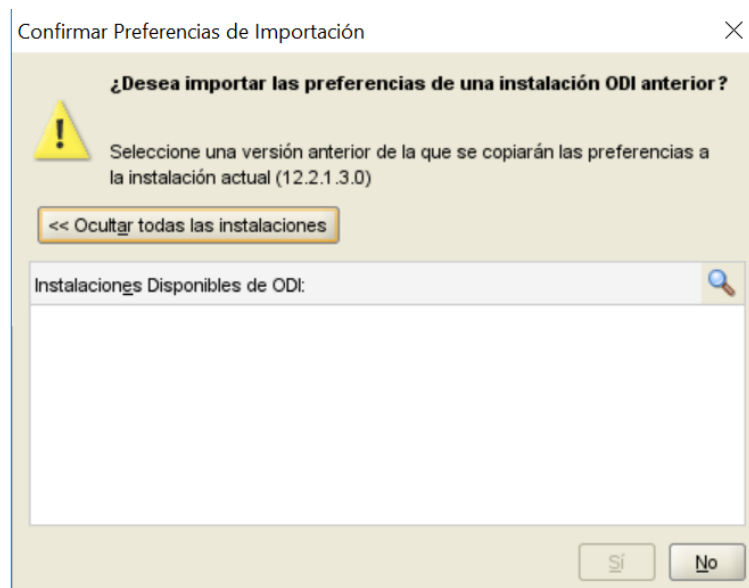
1. Quizá, al entrar por primera vez, nos pregunta la ubicación del JDK.



2. En mi caso, C:\Program Files (x86)\Java\jre1.8.0\_121 (Se puede consultar con Inicio; Programas; Java; Configurar Java; Ficha Java; Ver)



3. Quizá, nos aparece una pantalla para importar preferencias de instalaciones anteriores. Al ser la primera instalación de ODI, no hay instalaciones anteriores. Pulsamos el botón No para seguir.



4. Debemos crear la conexión al repositorio de trabajo, para ello necesitaremos especificar el usuario administrador de ODI Studio 12c, el usuario de base de datos del repositorio maestro y el usuario de base de datos del repositorio de desarrollo, con sus respectivas contraseñas



**Información de Conexión al Repositorio**

**Conexión a Oracle Data Integrator**

Nombre de la Conexión: DESODI

Usuario: SUPERVISOR

Contraseña: .....

**Conexión a Base de Datos (Repositorio Maestro)**

Usuario: DEV\_ODI\_REPO

Contraseña: .....

Lista de Controladores: Controlador JDBC de Oracle

Nombre del Controlador: oracle.jdbc.OracleDriver

URL: jdbc:oracle:thin:@localhost:1521:ord

**Repositorio de Trabajo**

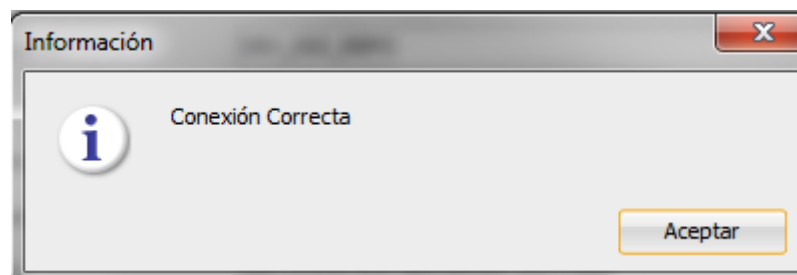
☐ Sólo Repositorio Maestro

☒ Repositorio de Trabajo DESREPO

☐ Conexión por Defecto

Ayuda Probar Aceptar Cancelar

5. Podremos probar si la configuración es correcta mediante el botón de Probar.



6. Si la conexión es correcta, pulsamos Aceptar, para guardar la conexión realizada.

7. Y Aceptar para conectar.

**Conexión de Oracle Data Integrator**

Nombre de la Conexión: DESODI

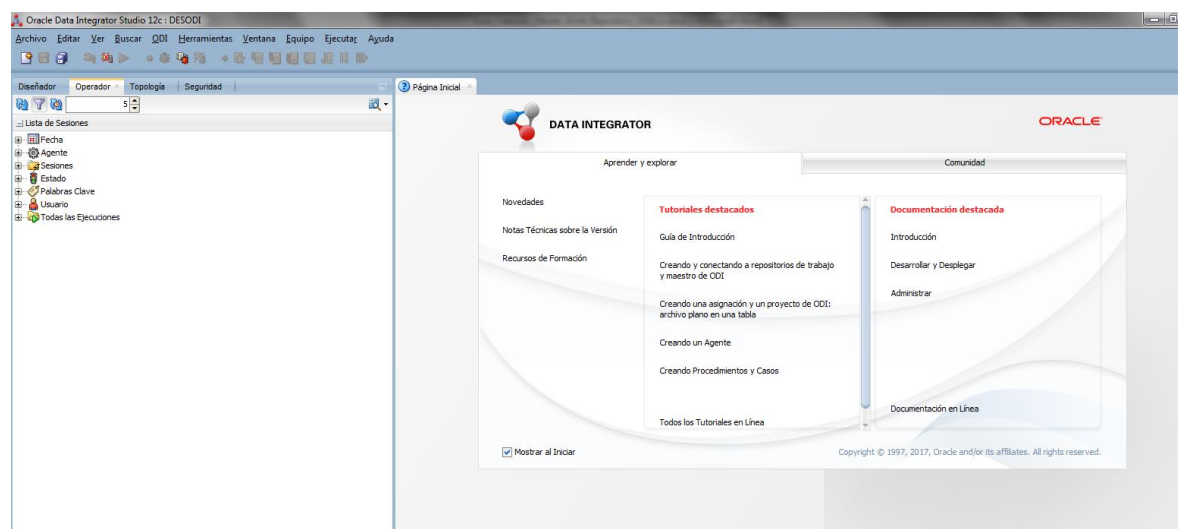
Usuario: SUPERVISOR

Contraseña: .....

Ayuda Aceptar Cancelar



Así ya nos habremos conectado a ODI Studio 12c y veremos una pantalla como la siguiente:



ODI Studio ofrece cuatro Navegadores para la gestión de los diferentes aspectos y fases de un desarrollo de un proyecto de integración:

### 1. Diseñador (Designer):

Se utiliza para el diseño y desarrollo de los procesos que intervienen en la integración de datos, ya sean ET-L's de carga de datos para análisis en los sistemas de Inteligencia de Negocio, o movimientos y transformación de datos entre diferentes sistemas operacionales o BigData en una organización.

- Ingeniería inversa de las aplicaciones existentes en las bases de datos.
- Desarrollo gráfico y mantenimiento de interfaces de transformación e integración.
- Visualización de los flujos de datos en las interfaces y sus dependencias

### 2. Operador (Operator):

Es la herramienta de monitorización de ejecuciones de interfaces y desarrollos. Además, se puede visualizar el estado de las ejecuciones con fecha de ejecución y duración.

### 3. Topologías (Topology):

Se utiliza para definir y administrar las arquitecturas físicas y lógicas de las fuentes o sistemas de información. La definición de las topologías consistirá en especificar las tecnologías, los tipos de datos, los servidores de datos relacionados con estas tecnologías y esquemas de los que contienen, los contextos y los agentes.

Los datos del servidor permitirán a ODI ejecutar las mismas interfaces de integración en diferentes entornos físicos.



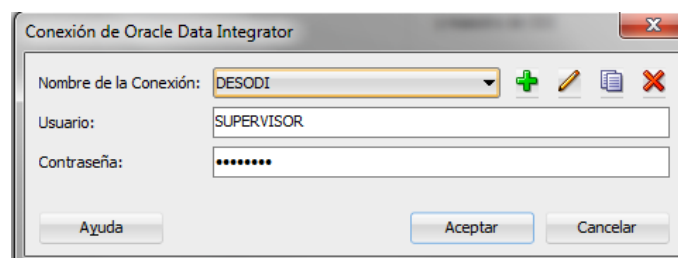
#### 4. Seguridad (Security)

Es la herramienta para la gestión de la información de seguridad en ODI se puede crear usuarios y perfiles, otorgar o revocar a los usuarios privilegios para realizar o no determinadas acciones (editar, borrar, etc) en los objetos genéricos (servidor de datos, tipos de datos, etc).

### Ejercicio 2.-Creación de Arquitectura Lógica y Física.

Este ejercicio es para aprender a realizar movimiento de datos entre una base de datos PostgreSQL y Oracle. Concretamente vamos a crear una asignación para mover los datos de la tabla MÉDICO de Postgresql (Origen) al esquema stage de la base de datos Oracle12c (destino).

#### 1.- Conexión al Repositorio de Desarrollo.



La ficha **Topología** de ODI Studio es la parte reservada al diseño de las **arquitecturas** que intervienen en la integración de datos.

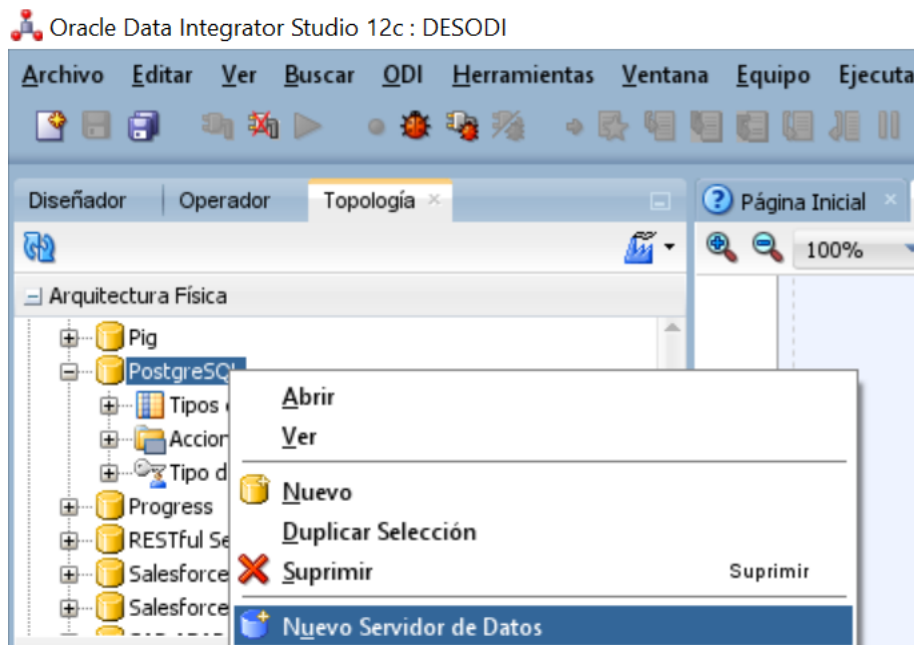
#### 2.- Creación de Arquitecturas Físicas en Topologías.

Desde la pestaña Topologías del ODI Studio. Situar en Arquitecturas Físicas.

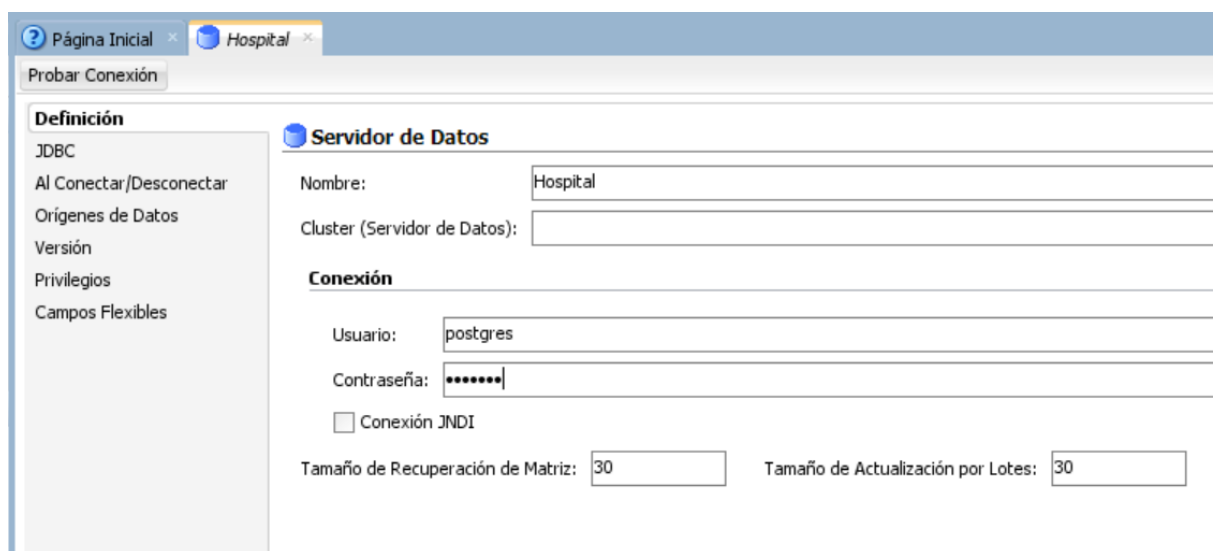
- a) Origen. Base de datos PostgreSQL

#### Servidor de Datos

Situarse en Tecnologías PostgreSQL. Aquí, botón derecho; Nuevo Servidor de Datos

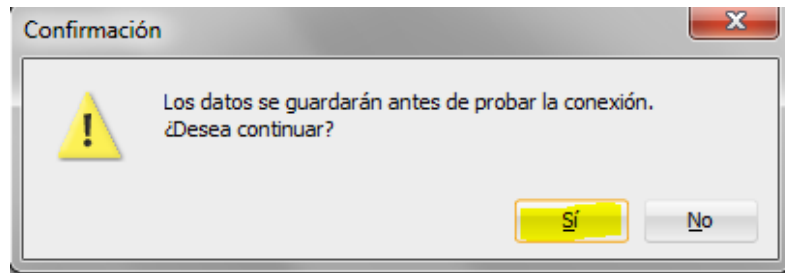


Se abrirá a la derecha una ventana donde especificaremos la base de datos **origen** de nuestros datos, concretamente el usuario postgres/postgres de conexión a la base de datos que hemos creado en la práctica y le llamaremos Hospital

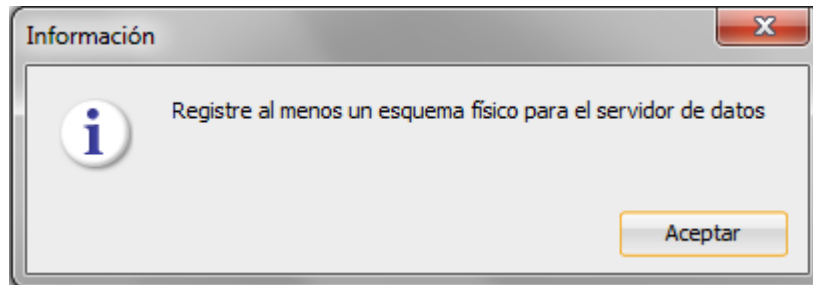


Pasaremos de la definición del servidor de datos, a la ficha de JDBC, para especificar los detalles de la cadena de conexión del origen de nuestros datos. En la parte de Controlador JDBC, En la URL, indicaremos el host, el puerto y la base de datos de conexión donde se encuentra el origen de los datos que estamos definiendo jdbc:postgresql://localhost:5432/postgres

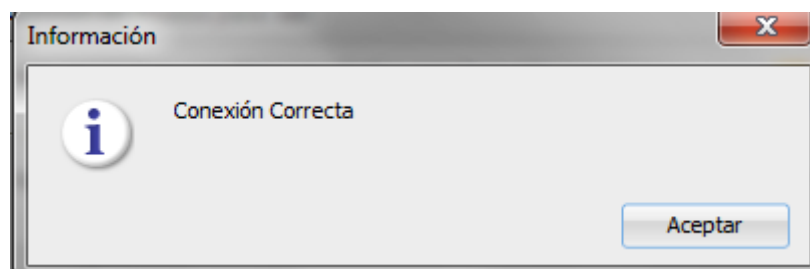
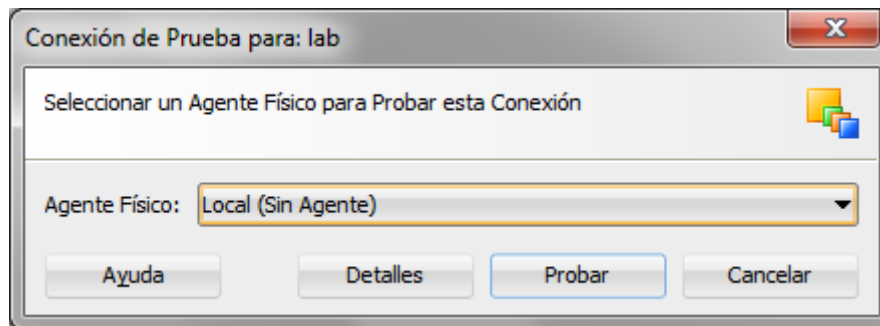
Pulsaremos al botón Probar Conexión, para comprobar que la configuración del servidor de datos origen es correcta. Antes de probar, tendremos que guardar la definición realizada del servidor de datos.



Nos saldrá una advertencia: Pulsaremos Aceptar



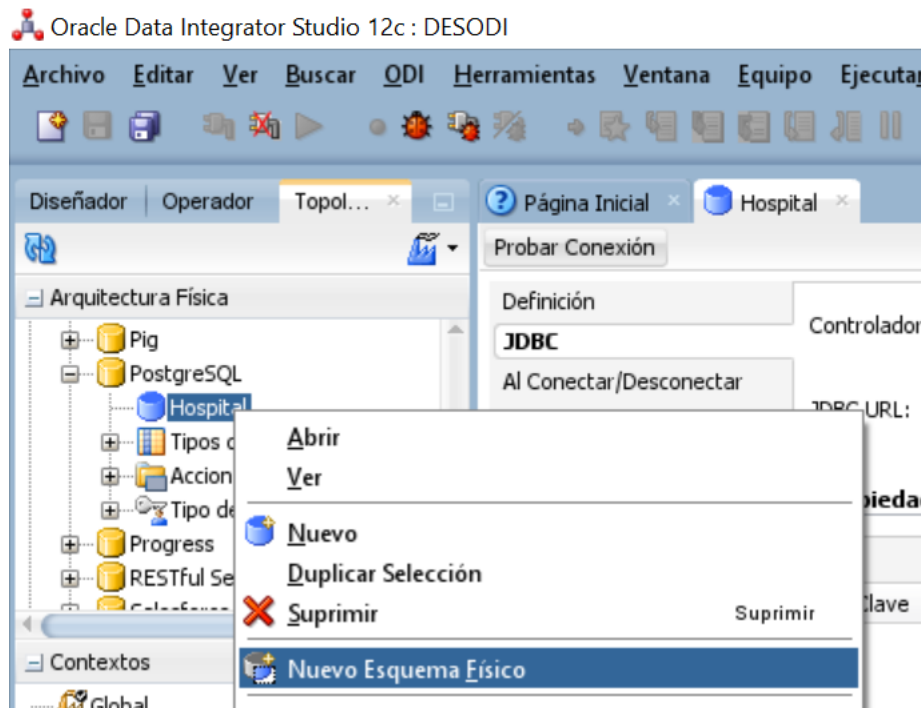
Para probar utilizaremos el agente local que nos muestra ODI proporcionado por la instalación de ODI Studio y pulsaremos Probar.



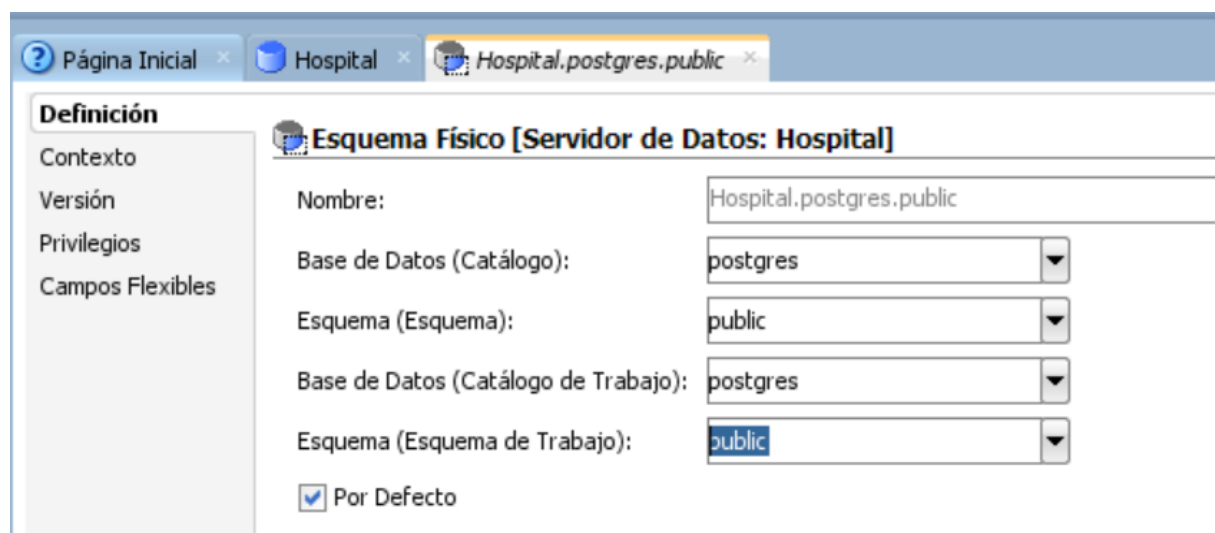
### **Esquema Físico**

Tal y como nos ha advertido ODI, para que la definición del servidor esté completa es necesario registrar un esquema físico de datos. Para ello, sobre el servidor de datos que acabamos de crear, Hospital, pulsamos botón derecho; Nuevo Esquema Físico.





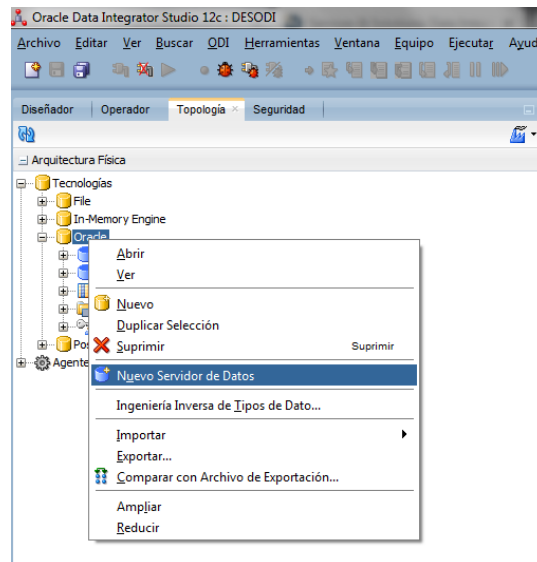
Se nos abre una ventana a la derecha, donde realizaremos la definición la base de datos y el esquema físico de datos. Elegiremos en los desplegables de Base de Datos **postgres** como bda origen de los datos y en los desplegables de Esquema, el esquema **public** dentro de la base de datos origen de los datos.



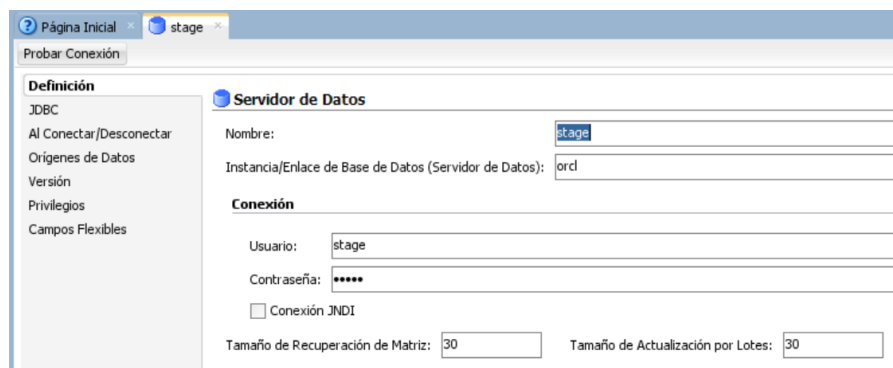
b) Destino. Esquema ORCL@lab

### Servidor de Datos

Situarse en Tecnologías Oracle. Aquí, botón derecho; Nuevo Servidor de Datos

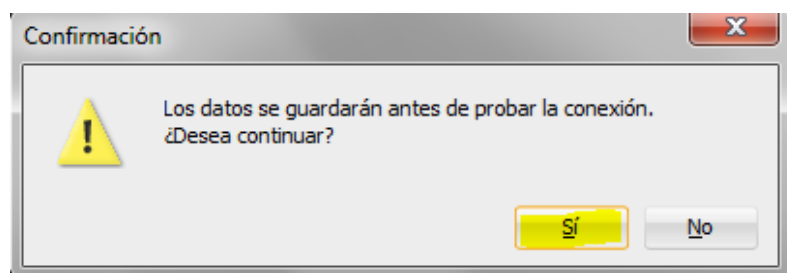


Se abrirá a la derecha una ventana donde especificaremos la base de datos **origen** de nuestros datos, concretamente el esquema stage/stage que hemos creado en la práctica de Base de Datos y le llamaremos lab

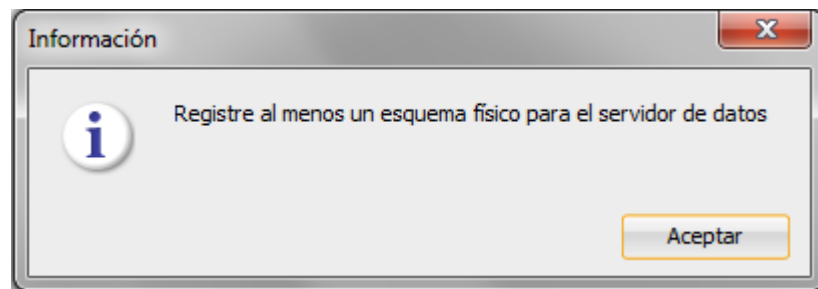


Pasaremos de la definición del servidor de datos, a la ficha de JDBC, para especificar los detalles de la cadena de conexión del origen de nuestros datos. En la parte de Controlador JDBC, Pulsaremos a la lupa de la derecha, elegiremos el driver oracle.jdbc.OracleDriver. En la URL, indicaremos el host, el puerto y la base de datos de conexión donde se encuentra el esquema **stage** origen de los datos que estamos definiendo.jdbc:oracle:thin:@localhost:1521:orcl

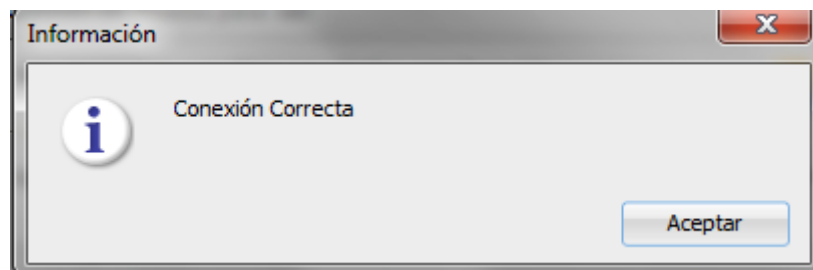
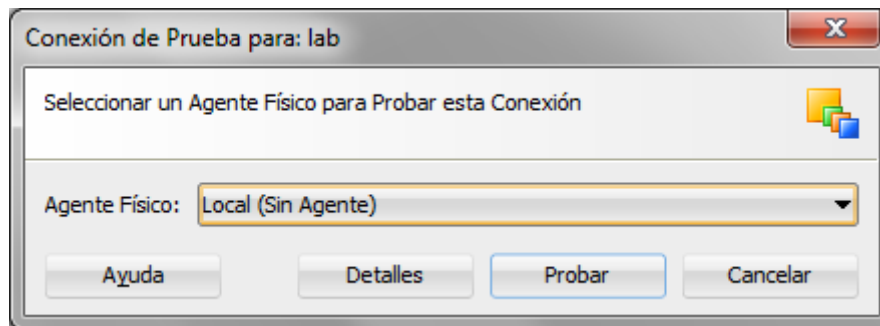
Pulsaremos al botón Probar Conexión, para comprobar que la configuración del servidor de datos origen es correcta. Antes de probar, tendremos que guardar la definición realizada del servidor de datos.



Nos saldrá una advertencia: Pulsaremos Aceptar

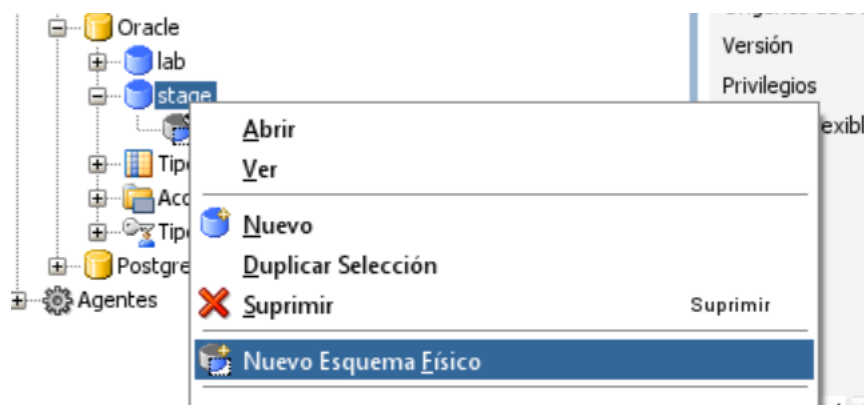


Para probar utilizaremos el agente local que nos muestra ODI proporcionado por la instalación de ODI Studio y pulsaremos Probar.



## Esquema Físico

Tal y como nos ha advertido ODI, para que la definición del servidor esté completa es necesario registrar un esquema físico de datos. Para ello, sobre el servidor de datos que acabamos de crear, stage, pulsamos botón derecho; Nuevo Esquema Físico.



Se nos abre una ventana a la derecha, donde realizaremos la definición del esquema físico de datos. Elegiremos en los desplegable de Esquema y Esquema de Trabajo el esquema de base de datos **stage** origen de los datos.



Definición

Contexto

Versión

Privilegios

Campos Flexibles

Esquema Físico [Servidor de Datos: stage]

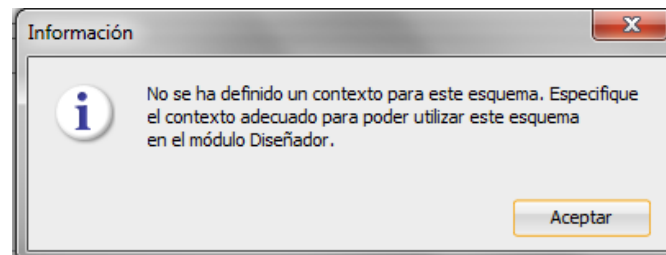
Nombre: stage.STAGE

Esquema (Esquema): STAGE

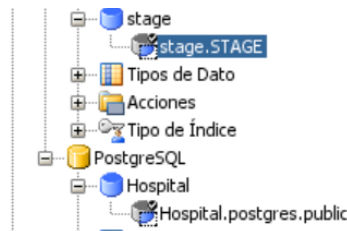
Esquema (Esquema de Trabajo): STAGE

☒ Por Defecto

Nos saldrá una advertencia: Pulsaremos Aceptar y Guardaremos la definición del esquema de datos.



Ya tenemos creadas las arquitecturas físicas origen y destino, con sus correspondientes servidores y esquemas de datos.



### 3.- Creación de contexto.

En ODI se utilizan los contextos para definir el entorno donde se quieren ejecutar los proyectos desarrollados.



En nuestro caso, la ejecución es local, así que crearemos un contexto que llamaremos local.

Definición

Agentes

Esquemas

Versión

Privilegios

Campos Flexibles

Contexto

Nombre: Local

Código: LOCAL

Contraseña:

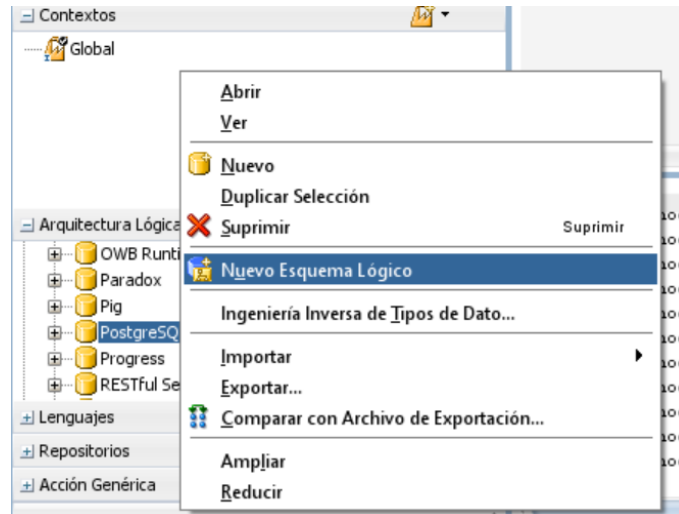
☐ Contexto por defecto utilizado por agentes



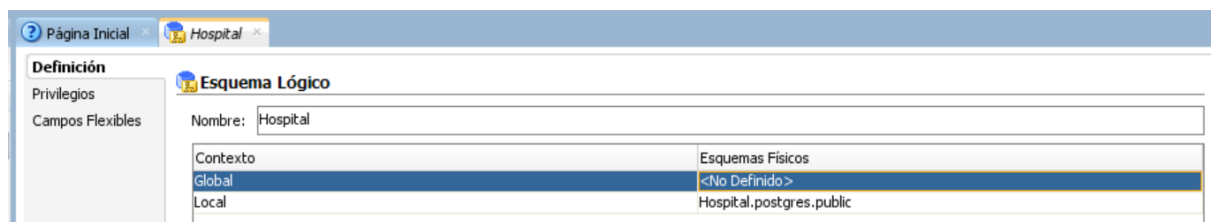
#### 4.- Creación de Arquitecturas Lógicas

Dentro de la ficha de ODI Studio Topologías, ir a Arquitectura Lógica

- Origen
- Situarse en Tecnologías Oracle. Aquí, botón derecho; Nuevo Esquema Lógico

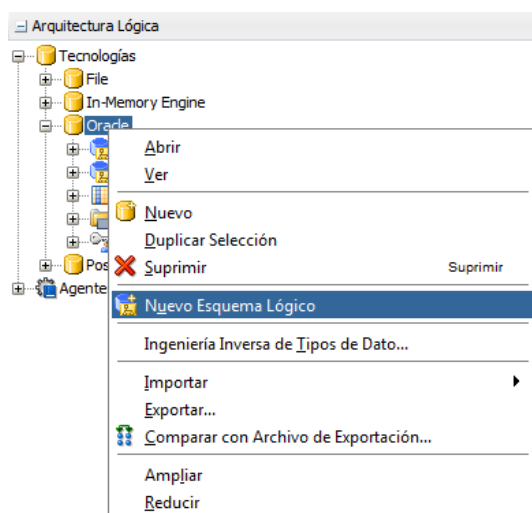


La definición de la arquitectura lógica, es simplemente darle un nombre, le llamaremos Hospital y relacionarlo con la arquitectura físico, mediante el contexto local. Cerrar y Guardar.



- Destino

Situarse en Tecnologías Oracle. Aquí, botón derecho; Nuevo Esquema Lógico

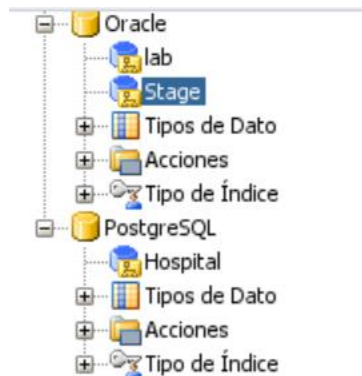




La definición de la arquitectura lógica, es simplemente darle un nombre, le llamaremos stage y relacionarlo con la arquitectura físico, mediante el contexto local. Cerrar y Guardar.

Definición	
Nombre: Stage	
Contexto	Esquemas Físicos
Global	<No Definido>
Local	stage.STAGE

Ya tenemos creadas las arquitecturas lógicas origen y destino.



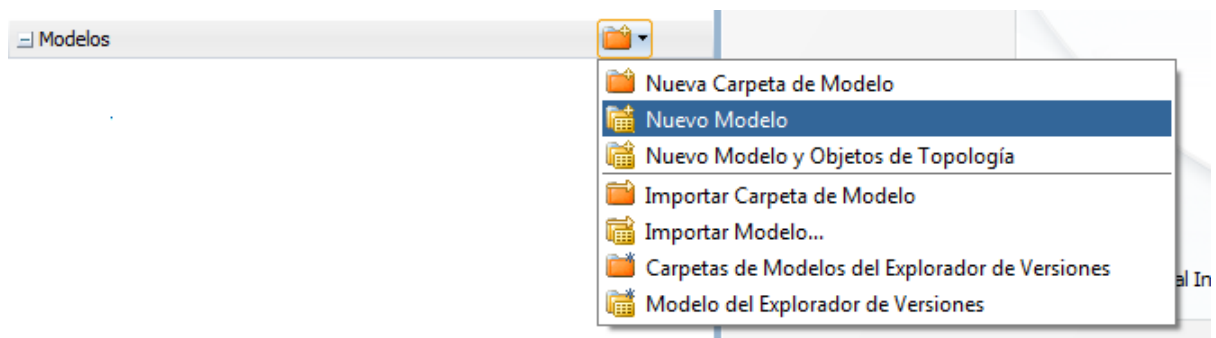
### Ejercicio 3.-Creación de Modelos de Datos.

La ficha **Diseñador** de ODI Studio es la parte reservada al desarrollo de los **procesos** que intervienen en la integración de datos, ya sean ET-L's de carga de datos para análisis en los sistemas de Inteligencia de Negocio, o movimientos y transformación de datos entre diferentes sistemas operacionales o BigData en una organización.

#### 1.- Creación de Modelos de Datos.

Los modelos de datos de ODI describen las estructuras de los datos, tanto los modelos de datos origen como los de destino, y siempre hacen referencia a las arquitecturas lógicas definidas en la topología.

##### a) Origen.



Se abre una nueva ventana a la derecha para la definición de nuestro modelo. Especificaremos nombre (hospital); Tecnología (PostgreSQL) Esquema Lógico, hace



referencia a la arquitectura lógica origen de los datos que se van a mover, en nuestro caso **hospital**

Definición

Ingeniería Inversa

Ingeniería Inversa Selectiva

Control

Registro en Diario

Tablas Registradas en Diario

Servicios

Marcadores

Modelo

Nombre: Hospital

Código: HOSPITAL

Tecnología: PostgreSQL

Esquema Lógico: Hospital

El paso siguiente es indicar la estructura de datos del modelo de datos “hospital”. Para ello utilizaremos la técnica de ingeniería inversa.

Para elegir las tablas que formarán parte del modelo de datos hospital, vamos a la ficha ingeniería inversa selectiva. Elegimos sólo las tablas de la base de datos postgres y hacemos clic en el botón de Ingeniería Inversa. Aceptamos la advertencia.

Definición

Ingeniería Inversa

**Ingeniería Inversa Selectiva**

Control

Registro en Diario

Tablas Registradas en Diario

Servicios

Marcadores

Memo

☒ Ingeniería Inversa Selectiva

☒ Nuevos Almacenes de Datos ☒ Almacenes de Datos Existentes ☒ Objetos para Ingeniería Inversa

Nombre de la Tabla	Tipo de Tabla
<input checked="" type="checkbox"/> medico	Tabla
<input checked="" type="checkbox"/> paciente	Tabla
<input checked="" type="checkbox"/> receta	Tabla
<input checked="" type="checkbox"/> visita	Tabla

Si cerramos la ventana de definición del modelo de datos hospital, veremos que se ha generado automáticamente su estructura.



Al ser una ingeniería inversa de una base de datos PostgreSQL, tendremos que revisar los tipos de datos de los atributos, para actualizarlos a tipos de datos Oracle, sino puede que los desarrollos fallen al ejecutar las cargas.

Como puede comprobarse debemos cambiar el tipo bpchar (no existe como tipo Oracle) a varchar2(100) en todos los campos importados.



Página Inicial | medico

Control Estático de Almacén de Datos

Definición

Atributos

Orden	Nombre	Tipo	Longitud	Escala	No Nulo	Comportamiento de SCD
1	num_med	integer		10	0	<No Definido>
2	nom_med	bpchar				<No Definido>
3	especialidad	bpchar				<No Definido>
4	direccion	bpchar				<No Definido>
5	ciudad	bpchar				<No Definido>
6	telefono	integer		10	0	<No Definido>
7	sueldo	integer		10	0	<No Definido>
8	dni	integer		10	0	<No Definido>

Los atributos actualizados quedarían así:

Página Inicial | Cargar\_Medico | medico

Control Estático de Almacén de Datos

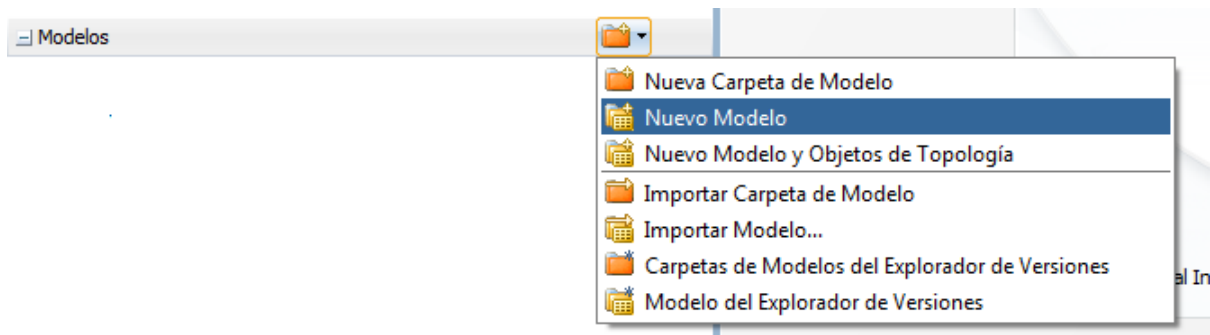
Definición

Atributos

Orden	Nombre	Tipo	Longitud	Escala	No Nulo	Comportamiento d...
1	num_med	integer		10	0	<No Definido>
2	nom_med	varchar	100			<No Definido>
3	especialidad	varchar	100			<No Definido>
4	direccion	varchar	100			<No Definido>
5	ciudad	varchar	100			<No Definido>
6	telefono	integer		10	0	<No Definido>
7	sueldo	integer		10	0	<No Definido>
8	dni	integer		10	0	<No Definido>

b) Destino.

Crearemos el modelo de datos origen. Desde el Diseñador, Modelos; Nuevo Modelo



Se abre una nueva ventana a la derecha para la definición de nuestro modelo. Especificaremos nombre (stage); Tecnología (Oracle) Esquema Lógico, hace referencia a la arquitectura lógica origen de los datos que se van a mover, en nuestro caso **stage**

Página Inicial | Stage | Stage

Ingeniería Inversa | Comprobar Modelo | Generar y Desplegar Servicios de Datos...

Definición

Ingeniería Inversa

Modelo

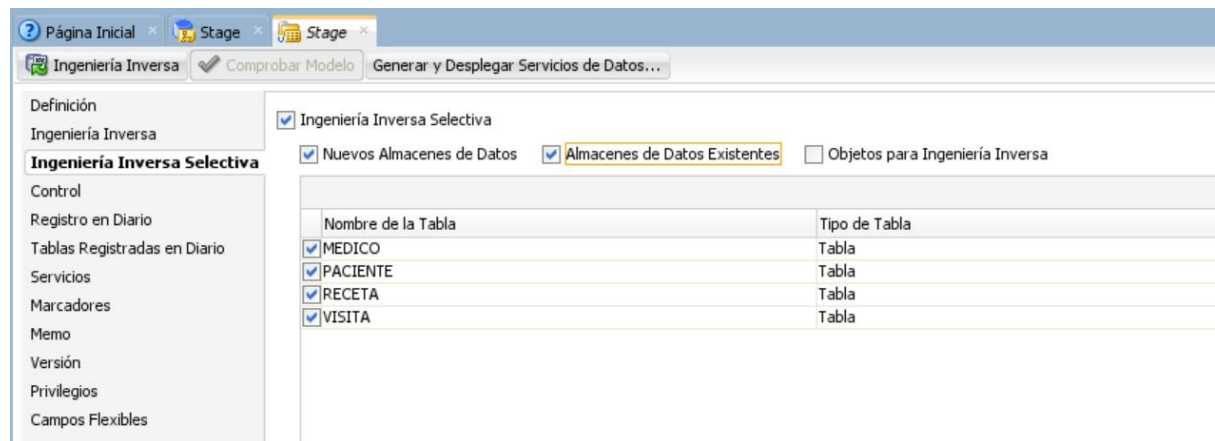
Nombre:	Stage
Código:	STAGE
Tecnología:	Oracle
Esquema Lógico:	Stage
Grupo de Acciones:	<Acción Genérica>
Carpeta por Defecto:	



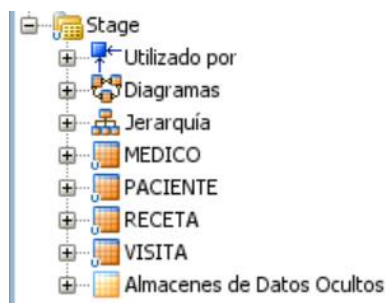


El paso siguiente es indicar la estructura de datos del modelo de datos “stage”. Para ello utilizaremos la técnica de ingeniería inversa.

Para elegir las tablas que formarán parte del modelo de datos stage, vamos a la ficha ingeniería inversa selectiva para elegir las tablas y hacemos clic en el botón de Ingeniería Inversa. Aceptamos la advertencia.

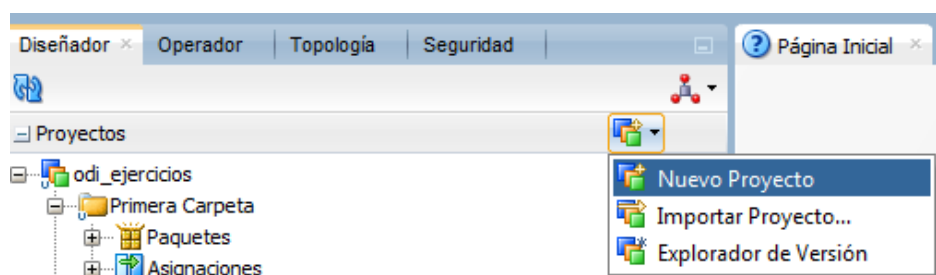


Si cerramos la ventana de definición del modelo de datos lab, veremos que se ha generado automáticamente su estructura.



## Ejercicio 4.-Creación de Asignación/Mapping.

Para crear un proceso de integración, lo primero es crear un proyecto. A un proyecto nuevo, ODI por defecto le crea una estructura de objetos. Para crear un proyecto, simplemente en la barra de herramientas, elegimos Nuevo Proyecto y aparece una ventana a la derecha para asignarle un nombre. Le llamaremos **odi\_ejercicios**.



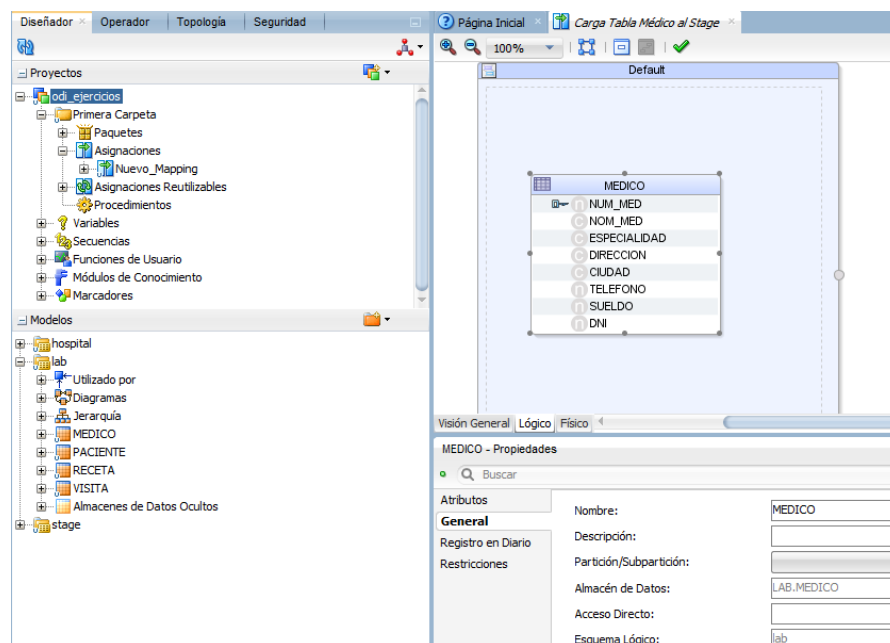
Ahora crearemos un mapping dentro del proyecto odi\_proyectos. En ODI12c los mappings son asignaciones. Vamos hacer una asignación para cargar la tabla **MEDICO** del modelo origen lab, al modelo destino stage.



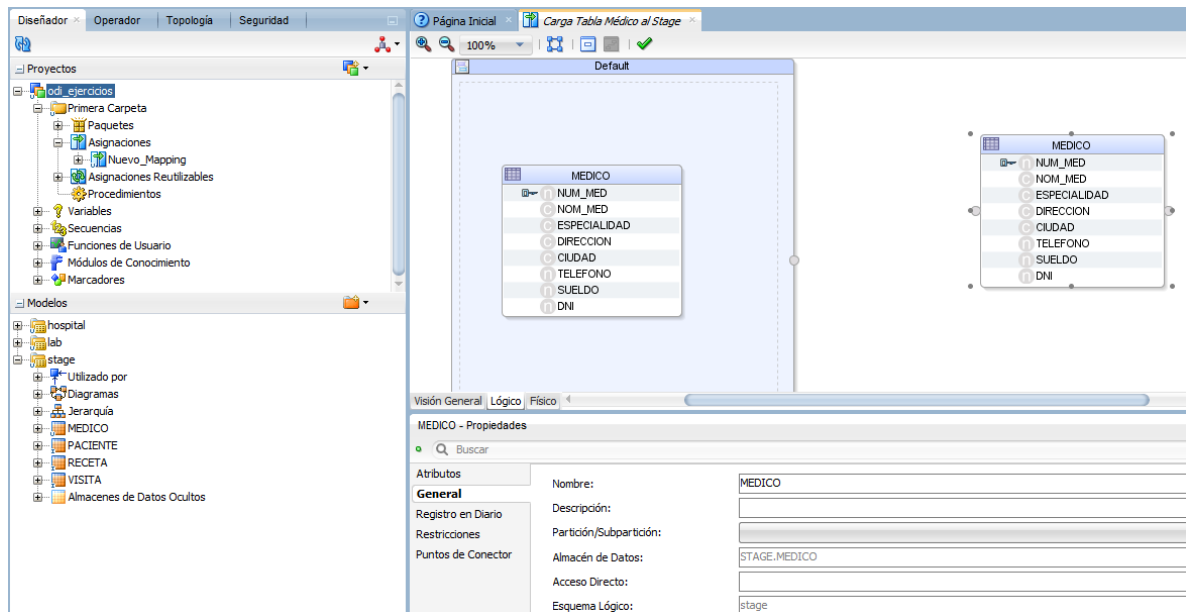
El concepto mapping/asignación, se refiere a un proceso que permite el movimiento de datos entre un origen y un destino. ODI es una herramienta con diseño declarativo, de esta forma la creación de mapping es gráfica, haciendo su diseño fácil e intuitivo.

Para crear una asignación, hacemos botón derecho en Asignaciones y en el menú contextual elegimos Nueva Asignación. Nos pedirá un nombre, le llamaremos Carga Tabla Medico al Stage, y pulsamos Aceptar.

El siguiente paso es indicarle a la asignación los modelos origen y destino. Para ello, lo haremos arrastrando la tabla MEDICO a cargar, sin soltar desde la estructura del modelo origen "hospital" en la ventana de Modelos, hasta la ventana del editor de asignaciones que vemos en pantalla. Soltamos el ratón en medio del área azul del cuadro Default. Tras la operación realizada, la asignación quedaría como la imagen siguiente:

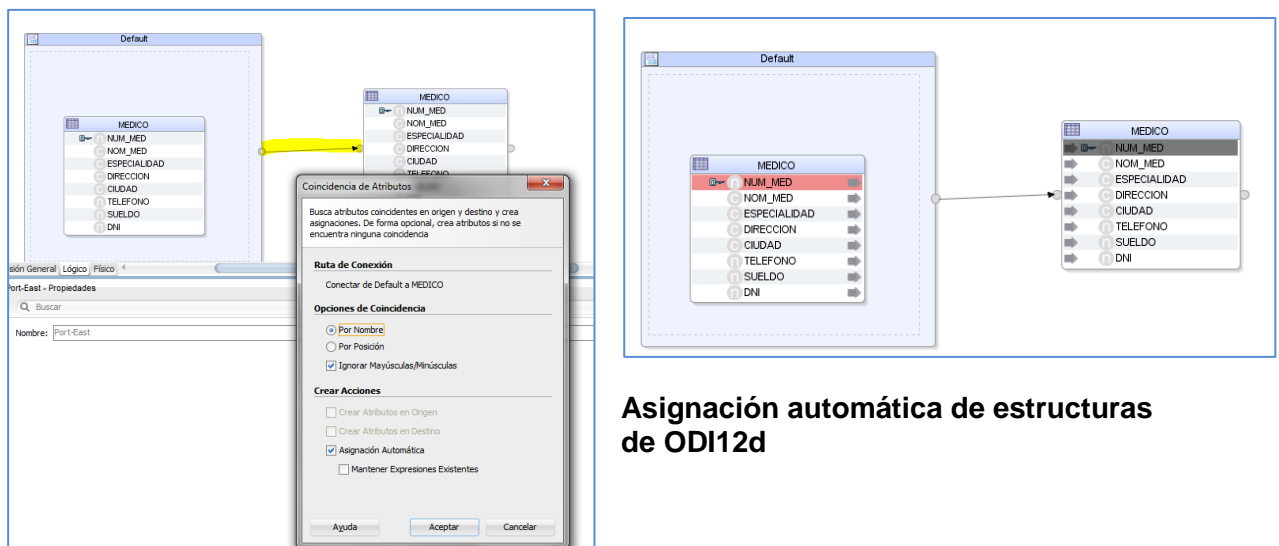


Ahora lo mismo para el modelo destino stage. La diferencia ahora es soltar fuera del área azul, el modelo destino lo debemos dejar en el área de color blanco. Hacer clic en la tabla MEDICO del modelo stage y arrastras sin soltar hasta dejarlo en la zona en blanco del editor de la asignación. Tras la operación realizada, la asignación quedaría como la imagen siguiente:




El siguiente paso es hacer la asignación propiamente dicha. La asignación consiste en dos pasos:

Primero indicar los pasos del flujo, simplemente es unir el primer paso con el siguiente. Se hace clic en el círculo pequeño en el borde de la línea derecha de la zona azul de la asignación (corresponde a la parte del origen de los datos del flujo de datos) y se **une, haciendo una flecha imaginaria**, con el círculo de la línea izquierda de la tabla destino. Tras la operación, ODI nos visualiza la ventana de asignación automática, pulsaremos Aceptar.



### Asignación automática de estructuras de ODI12d

Ahora sólo falta ejecutar el mapping para cargar los datos del origen al destino.

Para ejecutar pulsaremos el botón del triángulo verde  ODI nos solicitará guardar la asignación antes de ejecutar. Elegimos el contexto Local para la ejecución y pulsaremos Aceptar al resto de valores por defecto de los parámetros de ejecución.



Ejecutar

Contexto: Local

Diseño de Asignación Física: Físico

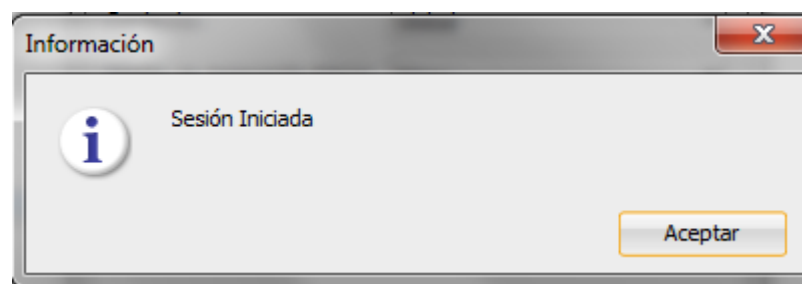
Agente Lógico: Local (Sin Agente)

Nivel de Log: 5

☐ Simulación

Ayuda Aceptar Cancelar

Aceptamos también la ventana de información de inicio del proceso de ejecución



El resultado de la ejecución del mapping se verá en la ficha Operador de ODI Studio. La ficha **Operador** es la parte reservada a la ejecución de los **procesos** que intervienen en la integración de datos. Si ha finalizado correctamente aparecerá un visto bueno.

Mediante SQLDeveloper comprobaremos que el destino stage, existe la tabla y contiene datos, exactamente los mismos que la tabla MEDICO.

NUM_MED	NOM_MED	ESPECIALIDAD	DIRECCION	CIUDAD
1	1 Severino Martin	Atención Primaria	(null)	Barcelona
2	2 Marisa Senin	Atención Primaria	Rambla Catalunya 27	Tarragona
3	3 Pawel Pikorski	Cardiología	Paseo Cracovia 23	Girona
4	4 Emilia Rodriguez	Radiología	El Pelin 189	Mataró
5	5 Ana Maria López	Traumatología	(null)	Barcelona
6	6 Conrado Pitarque	Cirugía Plástica	Las Hermosas 78-15a	Tarragona
7	7 Andrés Libertad	Atención Primaria	Avenida de las Marinas 2	Penedès
8	8 Celestino Pose	Odontología	(null)	Barcelona
9	9 Elena Rodriguez	Atención Primaria	Las Sombras 56	L'Hospitalet
10	10 Mateo Austriaco	Traumatología	Urb. Cancelas Apt 45	Premià de Mar
11	11 Minerva Guerra	Oncología	Las Azores 90-6C	Barcelona
12	12 Pedro Ramirez	Cardiología	C/Primera a la Derecha 3	Barcelona

## Ejercicio 5.-Creación de Mapping sin tablas en destino.

En ejercicio servirá para cargar la tabla MÉDICO del stage al esquema lab, donde no existe la tabla.

Para crear la tabla en el modelo lab, dado que no existe la tabla en el destino, copiaremos la tabla MEDICO del modelo stage. Con el menú contextual, Copia y Pegar.

Añadimos un atributo nuevo “CUENTA” en la tabla MÉDICO del modelo lab.



La creación del mapping es exactamente igual que en el ejercicio 4, pero vamos hacemos una transformación, añadiendo el valor por defecto 1 al atributo cuenta del modelo MEDICO.

Antes de ejecutar, debemos realizar cambios sobre el desarrollo predeterminado de ODI para adaptarlo a nuestras necesidades. Nuestro modelo físico en destino en realidad no existe, solo hemos definido lo que será su estructura de datos copiándola del modelo de datos origen.

Lo que debemos decir a ODI es que al ejecutar debe crear el modelo físico siguiendo la estructura de datos. Eso se lo indicamos modificando los valores por defecto de los módulos de Conocimiento.

Los Módulos de Conocimiento (ODI Knowledge Modules) son el núcleo de la arquitectura ODI. Estos módulos contienen el conocimiento requerido para desempeñar un set de tareas relacionadas a una tecnología. Existen diferentes módulos de conocimiento que se aplican en las diferentes partes de ejecución del proceso de integración. En nuestro ejercicio intervienen los de Carga (LKM) y de Integración (IKM).

Los tipos de KM:

ODI dispone de una librería de diversos módulos KM para adaptarlos a medida y beneficiarse con las mejores prácticas de ELT.

OBJETO	TIPO KM	
Interfaces	LKM	<u>Loading</u>
	IKM	<u>Integration</u>
Modelos	CKM	<u>Check</u>
	RKM	<u>Reverse Engineering</u>
	JKM	<u>Journalizing</u>
	SKM	<u>Data Services</u>

En nuestro caso, cambiaremos una propiedad del módulo de conocimiento de integración. El cambio se realiza en la definición física del proceso, hay que cambiar a la ficha Físico y hacer clic en tabla MEDICO. Se visualizará la ventana de las propiedades de la tabla. Hay que situarse en las propiedades del Módulo de Conocimiento de Integración, y cambiar la propiedad CREATE\_TARGET\_TABLE de False a True. Con esto estamos diciendo que se cree la tabla en el destino.



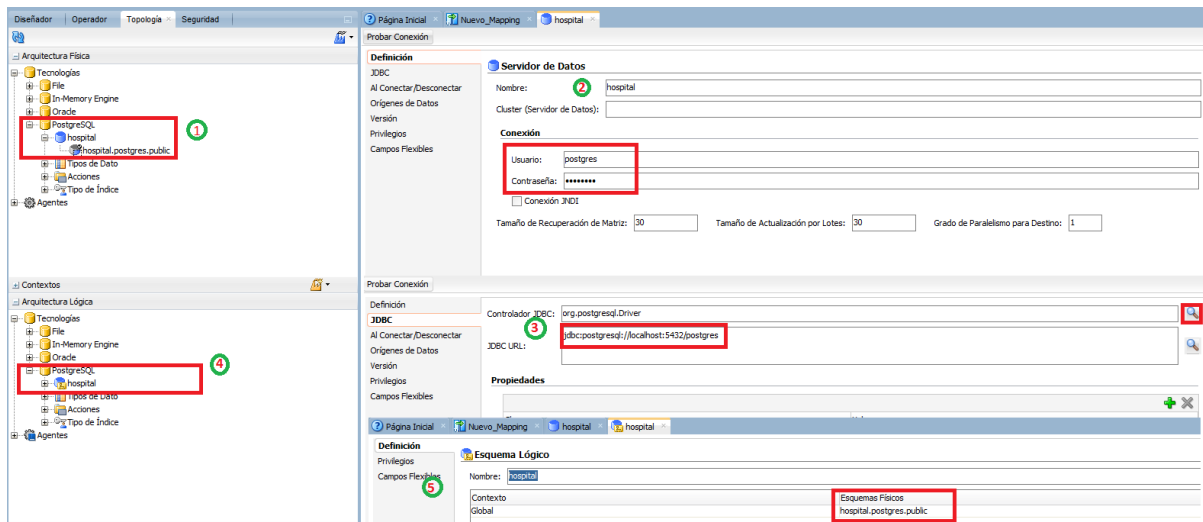
Nombre	Descripción	Ay...	Valor	Usar Valor p...
CREATE_TARGET_TABLE	Create the target table	?	False	<input checked="" type="checkbox"/>
INSERT_HINT	Oracle DML loading hint for lo...	?	True	<input checked="" type="checkbox"/>
SELECT_HINT	Oracle extraction hint for que...	?	False	<input checked="" type="checkbox"/>
CONSTRAINTS	Disable/enable constraints be...	?	None	<input checked="" type="checkbox"/>
TRUNCATE_TARGET_TABLE	Truncate the target table	?	False	<input checked="" type="checkbox"/>
DELETE_ALL	Delete all rows	?	False	<input checked="" type="checkbox"/>

El resultado de la ejecución del mapping se verá en la ficha Operador de ODI Studio. La ficha **Operador** es la parte reservada a la ejecución de los **procesos** que intervienen en la integración de datos. Si ha finalizado correctamente aparecerá un visto bueno.

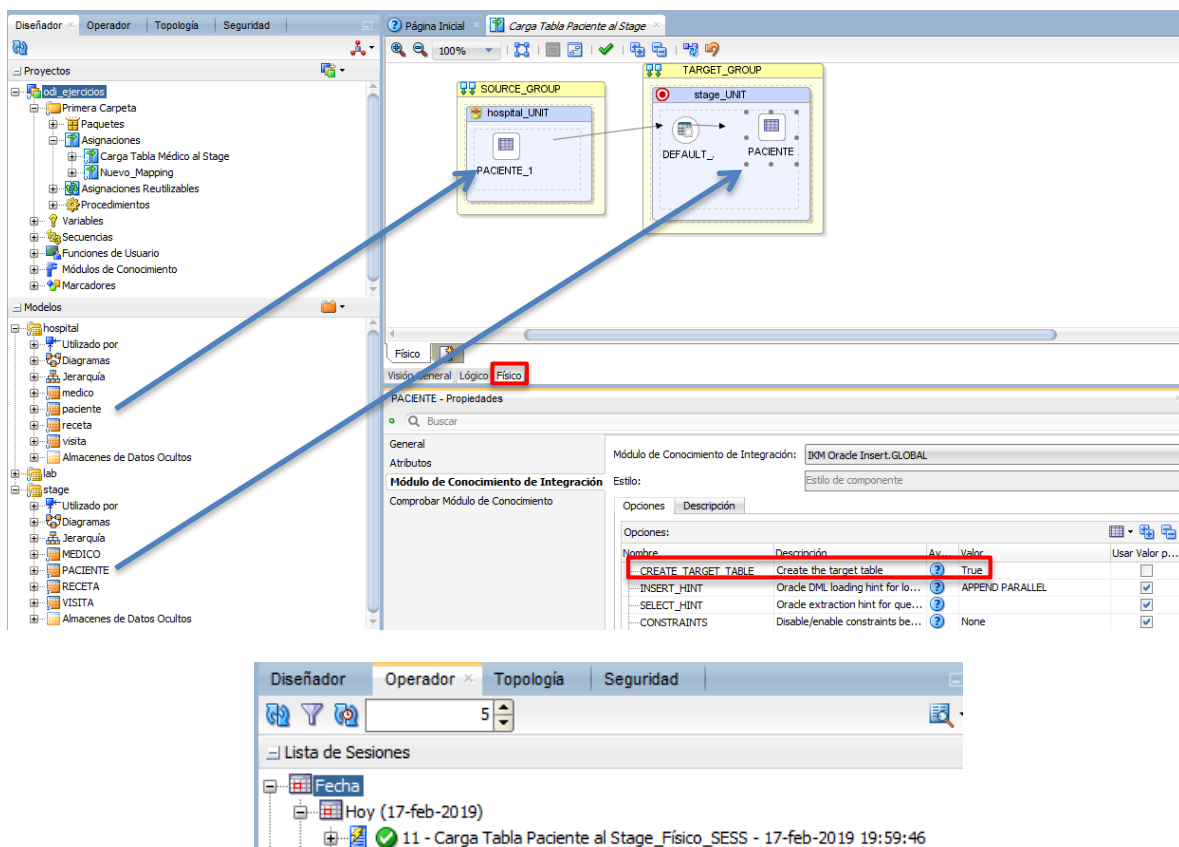
## Ejercicio 6.- Repaso.

Este ejercicio es para aprender a realizar movimiento de datos entre una base de datos postgresql y Oracle. Concretamente vamos a crear una asignación para mover los datos de la tabla PACIENTE de Postgresql (Origen) al esquema stage de la base de datos Oracle12c (destino).

- Crear arquitectura física origen PostgreSql
- Crear arquitectura lógica destino PosgreSQL



- Crear modelo de datos origen utilizando ingeniería inversa
- Crear asignación Cargar Tabla Paciente al Stage



## Ejercicio 7.-Creación de Paquetes.

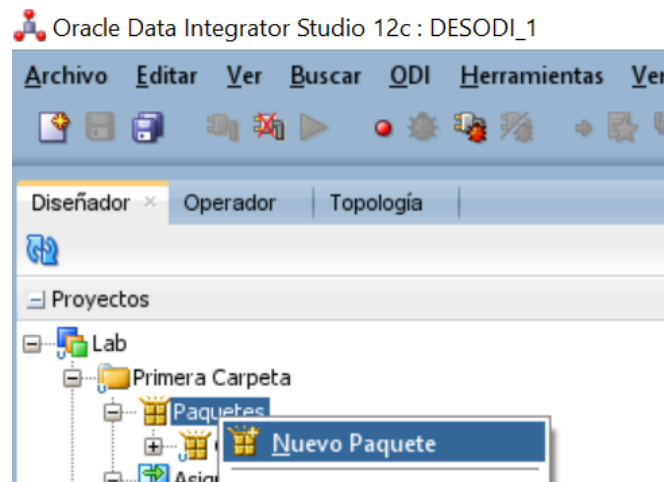
El componente de ODI que permite la ejecución de varios procesos desarrollados son los paquetes (packages)



El diseño de paquetes es muy fácil e intuitiva, ya que es la construcción de un workflow, compuesto de una secuencia de pasos enlazados. Dependiendo del resultado éxito o fallo en la ejecución de un paso, se deriva en la ejecución de otro.

Vamos a crear un paquete, Carga\_Tablas\_Stage, que consistirá en la ejecución de la carga de las tablas Médico y Paciente al Stage. En caso de error en la ejecución, se generará un sonido de error. Si todas las cargas se hacen correctamente, se generará un fichero de log, salida.txt.

Para crear el paquete, en Proyectos; vamos a la carpeta de Paquetes y elegimos Nuevo Paquete del menú contextual. Y le damos un nombre.



Para construir los pasos del diagrama de ejecución, vamos arrastrando al área de trabajo del paquete, los procesos que queremos que se ejecuten.

El primer paso del paquete, es la carga del mapping Carga\_Médico\_Stage, para ellos vamos a Asignaciones, y arrastramos la asignación hasta el área de trabajo. Haremos lo mismo, arrastrar y soltar, con la asignación para la carga de la tabla Paciente.

Una vez que tengamos los procesos que queremos que se ejecuten, lo que haremos es unir los pasos con las flechas de ejecución correcta o ejecución incorrecta.

#### Paso 1.

La secuencia para el paso 1 será: si la carga de la tabla médico es correcta (OK), entonces, que siga al siguiente paso y se ejecute la carga de la tabla paciente. En caso contrario, es decir error (KO), arrastramos de las operaciones de ODI, OdiBeep, al área de trabajo.

#### Paso 2.

La secuencia para el paso 2 será: si la carga de la tabla paciente es correcta (OK), entonces, que siga al siguiente paso y se genera el archivo log de salida de final con éxito, arrastramos al área de trabajo la acción ODI OutputFile, completando los parámetros.

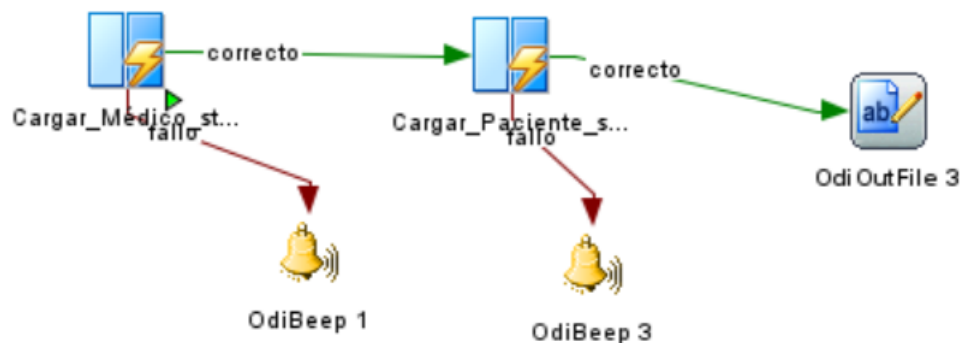




General	Comando	Avanzada	Memo	Versión	Privilegios														
<b>OdiOutFile -FILE (Obligatorio)</b>																			
Archivo de destino. Su ruta de acceso puede ser absoluta o relativa a la ubicación del agente de ejecución.																			
Nombre del Paso																			
OdiOutFile 3																			
<table border="1"><thead><tr><th>Parámetro</th><th>Valor</th></tr></thead><tbody><tr><td>Archivo de Destino</td><td>C:\Desktop\salida.txt</td></tr><tr><td>Agregar</td><td><input checked="" type="checkbox"/></td></tr><tr><td>Juego de Caracteres</td><td>ISO8859_1</td></tr><tr><td>Texto</td><td>"Termina con éxito"</td></tr><tr><td>Retorno de Carro</td><td></td></tr><tr><td>Esquema de Hadoop Lógico de Destino</td><td></td></tr></tbody></table>						Parámetro	Valor	Archivo de Destino	C:\Desktop\salida.txt	Agregar	<input checked="" type="checkbox"/>	Juego de Caracteres	ISO8859_1	Texto	"Termina con éxito"	Retorno de Carro		Esquema de Hadoop Lógico de Destino	
Parámetro	Valor																		
Archivo de Destino	C:\Desktop\salida.txt																		
Agregar	<input checked="" type="checkbox"/>																		
Juego de Caracteres	ISO8859_1																		
Texto	"Termina con éxito"																		
Retorno de Carro																			
Esquema de Hadoop Lógico de Destino																			

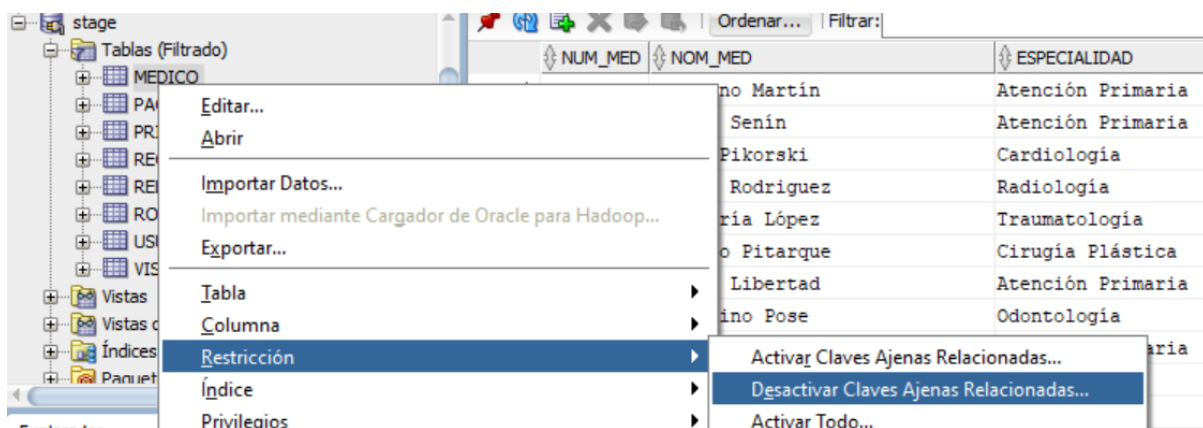
En caso contrario, es decir erro (KO), arrastramos de las operaciones de ODI, OdiBeep, al área de trabajo.

El diagrama de ejecución del paquete Cargas\_Tablas\_Stage completo, quedaría así:



Si existen datos en las tablas, antes hay que truncarlas. Para ello:

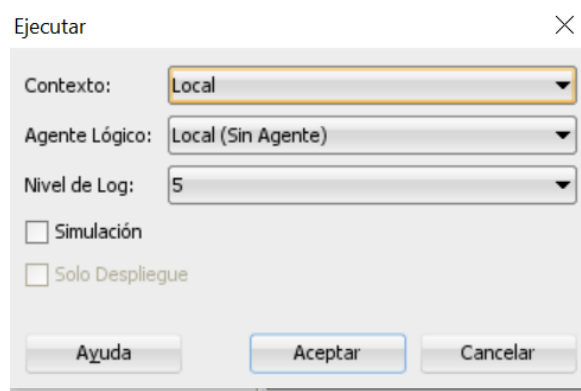
1.- En el SQLDeveloper, desactivamos las restricciones de las claves ajenas:



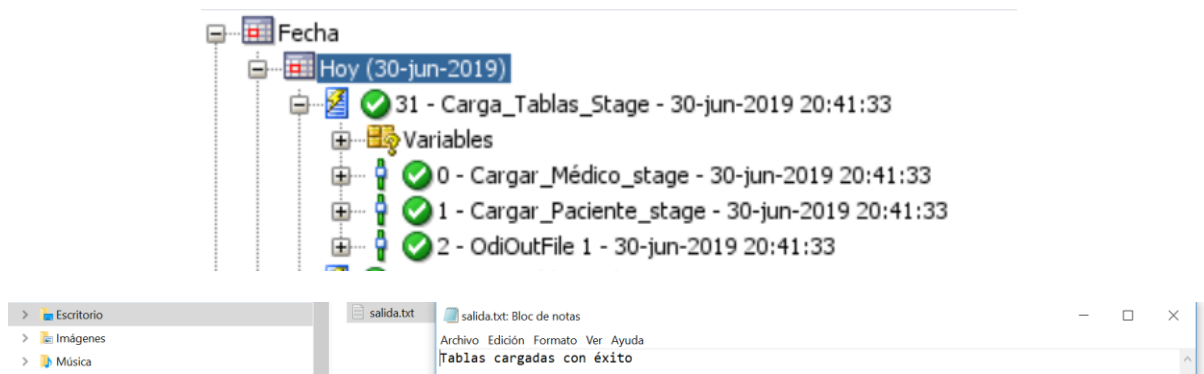
Y luego truncamos los datos con la sql: TRUNCATE TABLE MEDICO;

Haremos lo mismo si existen datos en la tabla PACIENTE.

Y ahora ya podemos ejecutamos el paquete en contexto Local con el agente local,



Y el resultado es la ejecución de la secuencia completa del paquete, con los tres pasos diseñados. Paso\_1. Carga de la tabla Medico al Stage, Paso 2. Carga de la tabla Paciente al Stage y Paso 3. Generación del fichero salidad.txt



Al terminar estos ejercicios habrás aprendido a definir arquitecturas, modelos de datos y procesos y paquetes de integración para el movimiento de datos entre esquemas de bases de datos.