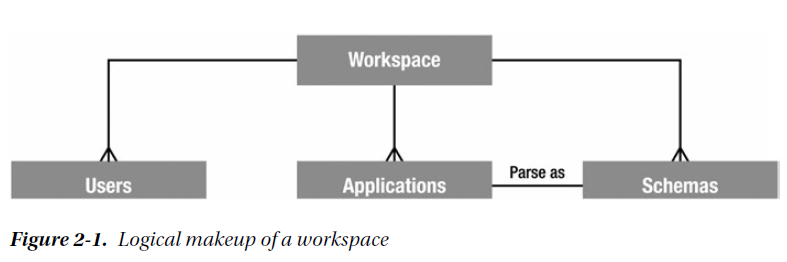
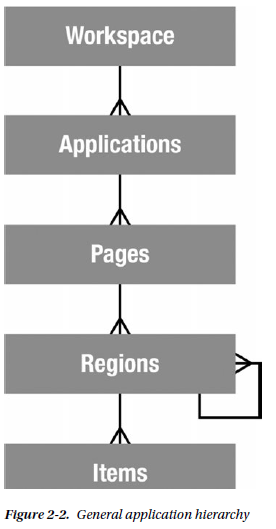
**Oracle APEX.**

**Libro: Begining Oracle Aplication Express 5**



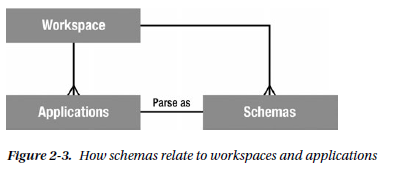
APEX 5.0 introduces the ability to use an external repository, such as Single Sign-on or LDAP, as a source

to assign and validate APEX users, meaning that a single user could have access to multiple workspaces.



*Regions* are UI items that serve as content containers. You can have any number of regions on a page, and regions can be nested in other regions. This gives you the opportunity to create things like dashboards,where you might nest a data report region and a graph region in a single parent HTML region.

***Items***are the HTML form elements that are used to present the UI to the user. These include things such as buttons, select lists, text fields, check boxes, radio groups, and so on. There are two categories of items: page-level items and **application-level items**. The difference is that the latter are defined at the application level and aren’t rendered directly on the page. You can think of these as **global variables**. Page-level items are defined on a specific page and are assigned to a region in order to control where and how they display to the user.



When a workspace is created, it’s linked with at least one, and possibly many, underlying database schemas. This provides access to database objects such as tables, views, stored PL/SQL program units, and so on.

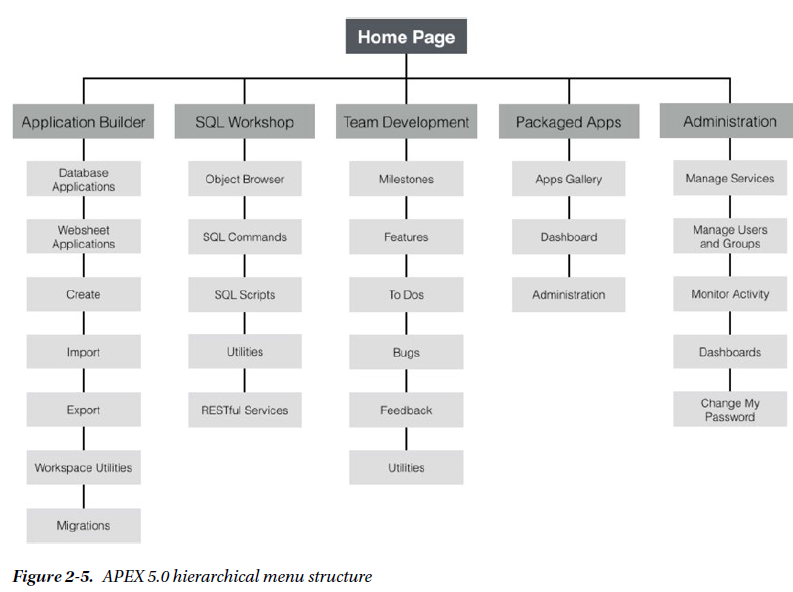
When an application is created, it’s assigned a single “parse as” schema from the list of schemas associated with the workspace. A *“parse as” schema* is the Oracle database user in which all SQL queries and PL/SQL calls run by that application are executed. So, if your application was defined with a “parse as” **schema of DOUG**, a query such as

**SELECT \* FROM EMP**

would execute in the database as if it were written

**SELECT \* FROM DOUG.EMP**

The **#OWNER#** replacement variable is substituted for the actual “parse as” schema for the application at runtime. So the statement



**SELECT \* FROM #OWNER#.EMP**

**The Page Designer**

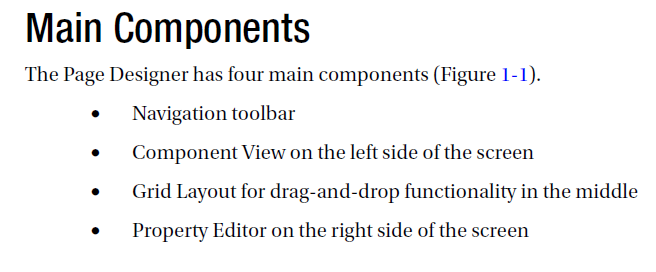
The Page Designer is where you’ll be spending most of your time as a developer creating and editing pages,

regions, and items.

**SQL Workshop**

The SQL Workshop is a suite of tools that provides developers with the ability to view and manage database

objects in the underlying schema(s) assigned to the workspace.



Variable que hace referencia al schema #OWNER#

Las consultas se hacen apuntando al schema.

Select \* from #OWNER#.emp

Lo cual se resuelve como

Select \* from BANESCO\_KPI.emp (nombre del schema)

BANESCO\_KPI es el “PARSE AS” SCHEMA de la aplicación.(solo puede ser uno)

IR: Interactive Report.

By default, all SQL statements executed via SQL Commands interface are automatically committed.

Pag. 37 Identifying the Problem and Desingning the Solution.

In APEX 4, the ability to use ROWIDs in place of primary keys was introduced to help solve the problem of multicolumn primary keys.

As a general rule of thumb, logic that controls or manipulates the UI is best placed in APEX, and logic that implements business rules or controls the data is best placed in stored program units in the database.

The APEX wizards make heavy use of database metadata for the objects in the “parse as” schema.

There are dozens of ways to draw ERDs, from pen and paper to high-end database-design tools.

However, I tend to take the middle ground and use Oracle’s SQL Developer Data Modeler.

**SQL Workshop / Object Browser : Crear las tablas del modelo.**

trigger named BI\_TICKETS (BI quiere decir Before Insert)

SQL Workshop’s Data Workshop utility to load and unload data from an Oracle schema in a number of ways.

Loading Data with the Data Workshop Utility

Lookup Table: Tablas de búsqueda (ejemplo caso del estatus:ABIERTO, CERRADO, PENDIENTE)

APEX can create a lookup table—complete with its own sequence, trigger, and foreign key—and modify the original table so it points to the new lookup table.

Pag. 60

The **SQL Scripts** tool of **SQL Workshop** allows you to create, upload, manage, and run SQL scripts.

**CREATE OR REPLACE FUNCTION** get\_status

(p\_status IN VARCHAR2)

**RETURN NUMBER**

**IS**

l\_status\_id status\_lookup.status\_id%TYPE;

**BEGIN**

SELECT status\_id INTO l\_status\_id FROM status\_lookup

WHERE status = p\_status;

RETURN l\_status\_id;

**END** get\_status;

/

Understanding User Interface Defaults

The Attribute Dictionary allows you to create more-generic UI Defaults based on attribute names.

Consider this a more macro-level definition. Attribute Dictionary definitions can also be assigned synonyms, allowing more than one attribute name to share the same actual definition.

Here are some things to note about UI Defaults:

Table Dictionary defaults always override Attribute Dictionary defaults.

When an item is created using UI Defaults, no relationship is established with the UI Default. Therefore, if you later change the definition of the UI Default, the changes aren’t propagated to previously created items.

Items created before UI Defaults have been established don’t inherit properties of the UI Default.

Developers can choose not to use UI Defaults, and even if they’re used, can override them after the component is created.

**UI Defaults** can be managed either from SQL Workshop’s Object Browser or from SQL Workshop’s Utilities

page.

The first step in **creating UI Defaults** is to synchronize the Table Dictionary with the database so it knows what tables are in your schema.

**Applications and Navigation**

Lists of Values (LOVs)

Applications in APEX are created through application imports, by copying an existing application, or

by running the Create Application wizard.

Las aplicacones “sample” vienen desbloquedas, miestras que las que no dicen “sample” están bloquedas y hay que desbloquearlas via el botón “Manage”.

Pag. 73 Applications from Scratch

Pag. 90 Public Pages

If any of the pages in an application require authentication, an appropriate authentication scheme must be applied to the whole application.

APEX lets you define individual pages as Public or Requires Authentication using a defining property of the page.

Public pages are useful for introductory landing pages, login pages, and information pages.

Navigation Bar Entries.

Placement of navigation bars is dictated by the page template substitution variable #NAVIGATION\_BAR#.

Global Pages

A *Global Page* is a special type of page that acts as a “master page” for your application and can be added

one per user- interface type (that is, you may have one Global Page for the Desktop UI and another for the

Mobile UI).

Lists

As briefly mentioned earlier in this chapter, the new Universal Theme uses static lists instead of Tabs for

navigation.

A list, as a shared component (unless it is designated as the default navigation list for the UI), doesn’t

display in an application directly. Normally, a list region must be configured on a page in order for the list to

be seen by the user. **APEX has a template type defined specifically to support lists.**

Lists of Values pag. 102

STATIC: Entries are automatically alphabetized.

STATIC2: Entries render in the order in which they’re entered.

The syntax for specifying a static LOV is as follows:

TYPE:DISPLAY;RETURN,DISPLAY;RETURN,...

The TYPE may be either STATIC or STATIC2.

Ejemplo: STATIC:C;1,A;2,D;3,B;4,

Form on a Table

Modify a form on a Table. Pag 115

oracle apex (en la nube)

workspace name: banesco\_kpi

user: ejimenmail@gmail.com

pass: salvador1208

Base de Datos Oracle:

user:SYSTEM

pass:salud

puerto:1521

SID:XE

Crear en las variables de ambiente:

ORACLE\_SID: XE

Crear directorio:

C:\apex\_salud

SQLPLUS>

//crear el tablaspace para la base de datos..

create tablespace appx datafile 'c:\apex\_salud\apex.dbf' size 2000M extent management local segment space management auto;

create temporary tablespace temp0\_02 tempfile 'c:\apex\_salud\temp.dbf' size 2000M;

SQLPLUS> exit

>Copiar apex en C:\apex\_salud

“Entrar a sqlplus”

>sys as sysdba

oracle

SQLPLUS>@apexins.sql appx appx temp0\_02 /i/

“Entrar a sqlplus”

>sys as sysdba

oracle

SQLPLUS>@apex\_epg\_config.sql c:\apex\_salud (donde estan los dbf y lo que hace es crear un directorio de imágenes)

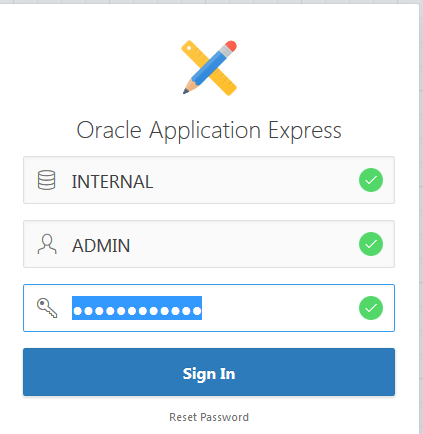
SQLPLUS>@apxldimg.sql c:\apex\_salud

SQLPLUS>@apxchpwd.sql (cambiar la contraseña)

user: ADMIN

password: Banesco#2017

http://localhost:8080/apex

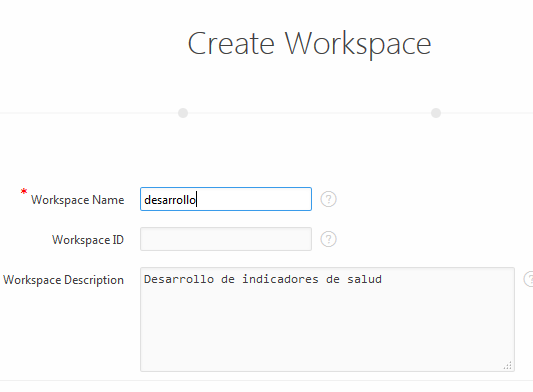


workspace defaul INTERNAL

usuario: ADMIN

passw: (por definir)

Crear workspaces:



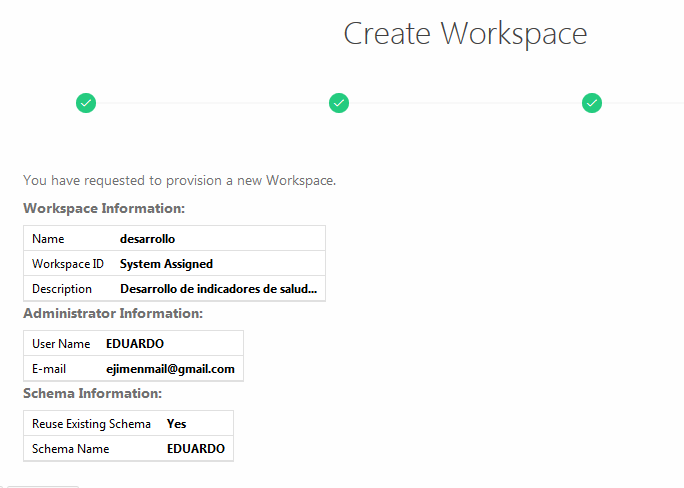
**CREAR UN ESQUEMA DE BD.**

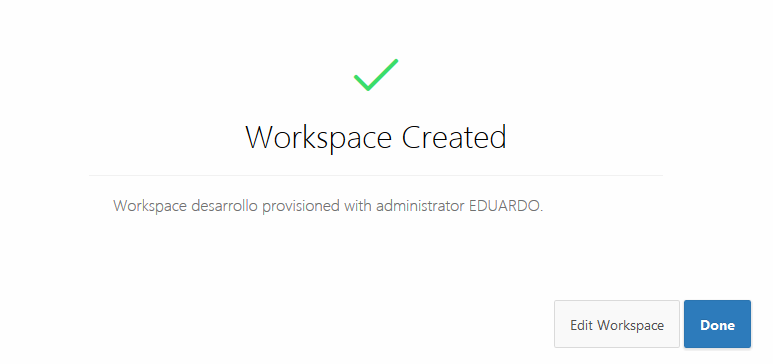
SQLPLUS> create tablespace APEX\_DEV datafile 'C:\oraclexe\app\oracle\oradata\XE\desarrollo.dbf' size 2000M extent management local segment space management auto;

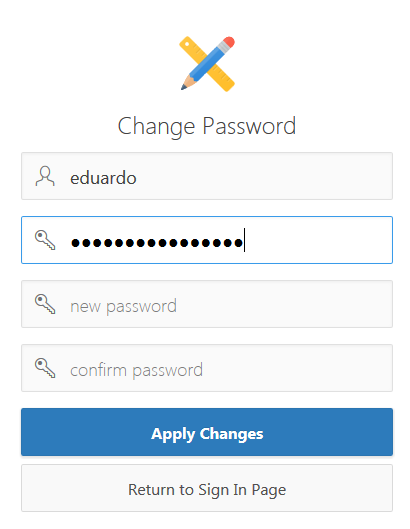
SQLPLUS> create user **eduardo** identified by **banescoSalud2017** default tablespace APEX\_DEV;

SQLPLUS> GRANT ALL PRIVILEGES TO EDUARDO;

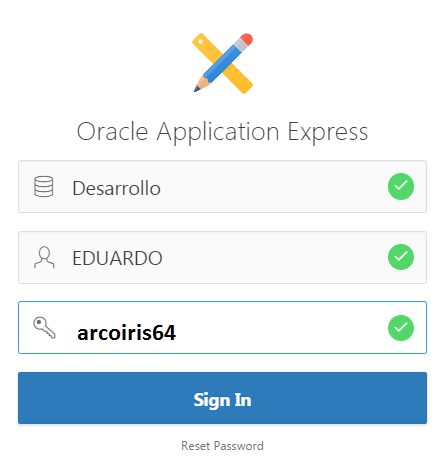
**CREAR EL WORKSPACE EN APEX.**







**NUEVO PASSWORD: arcoiris64**

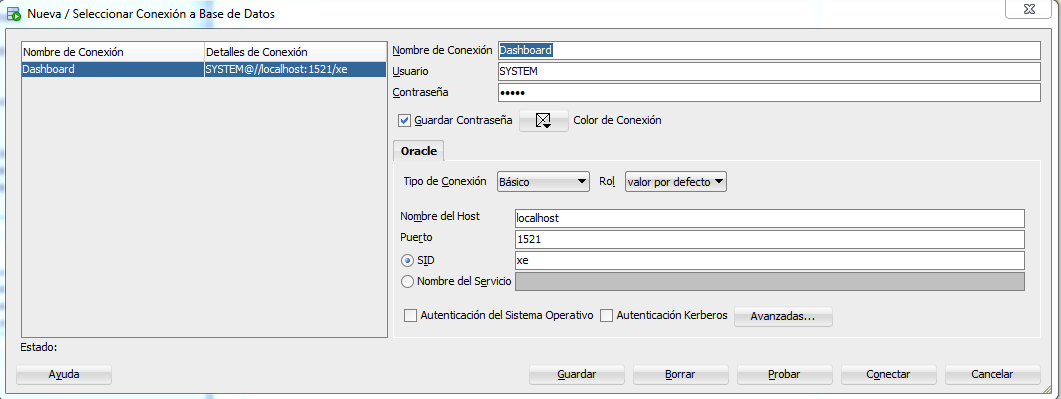


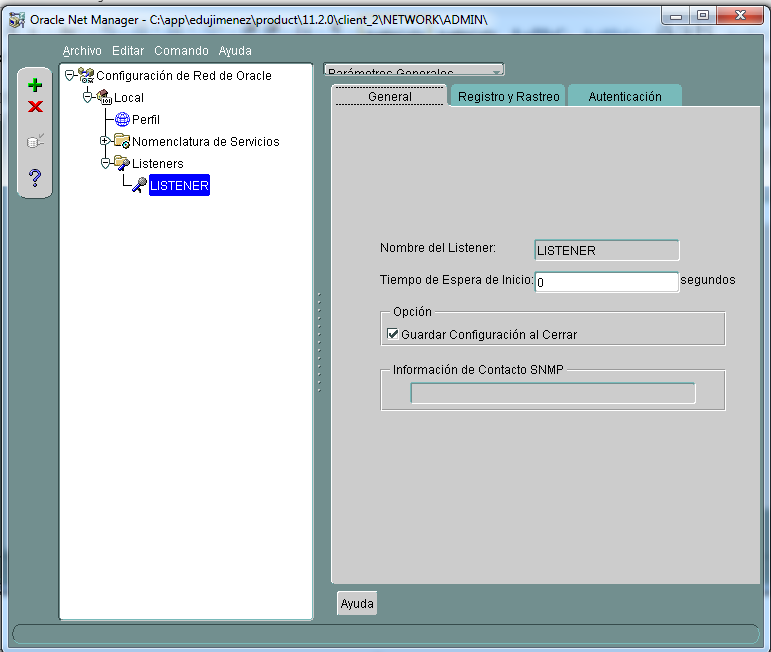
**oracle-apex-español.com (blog para las dudas)**

**Ubicaciones de los dbf en localhost: C:\oraclexe\app\oracle\oradata\XE**

create user salud identified by banescoSalud2017 Default tablespace APEX\_DEV;

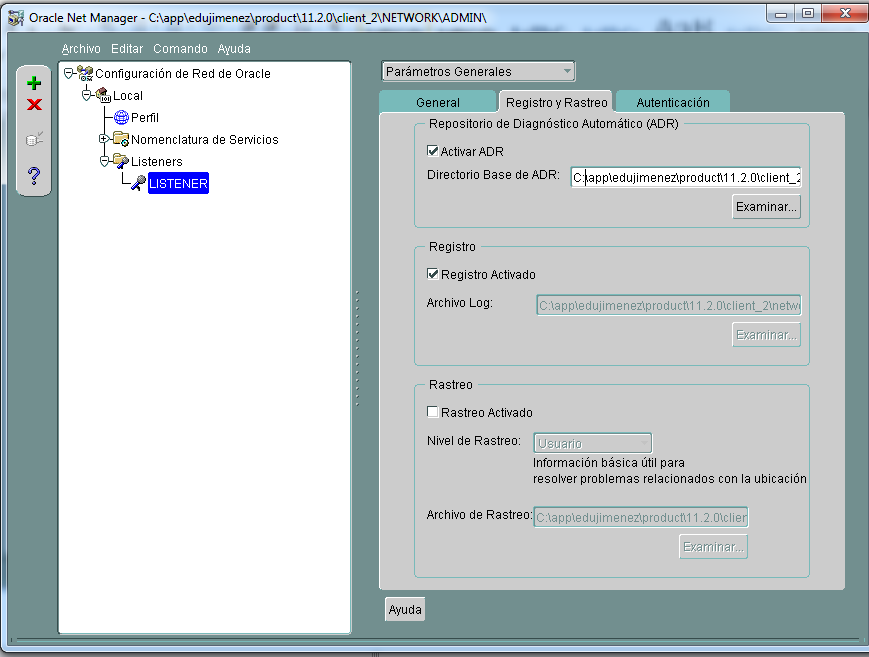
GRANT ALL PRIVILEGES TO EDUARDO;

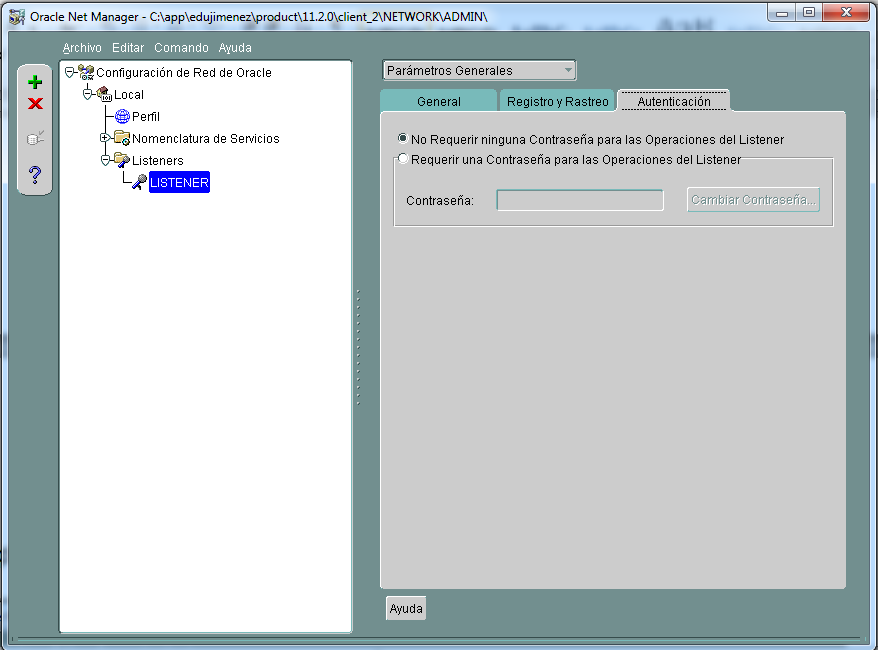


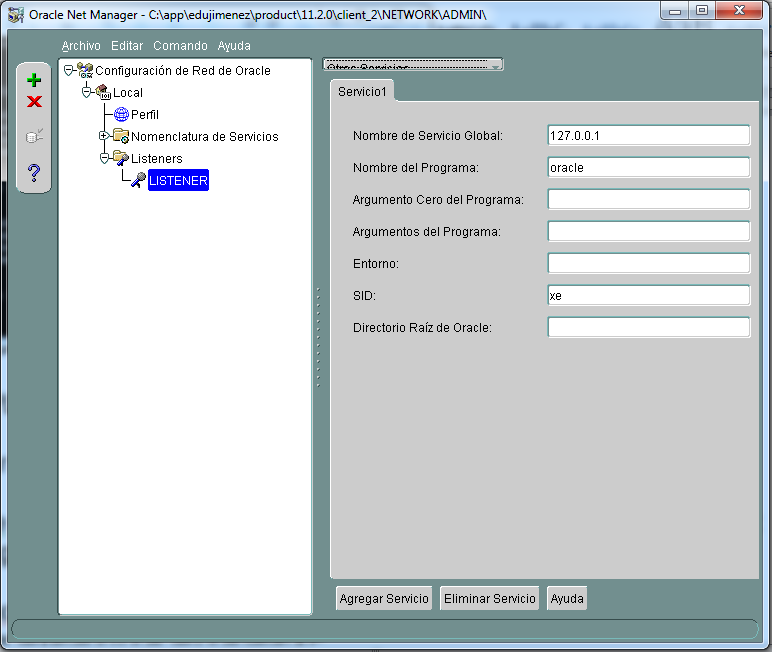


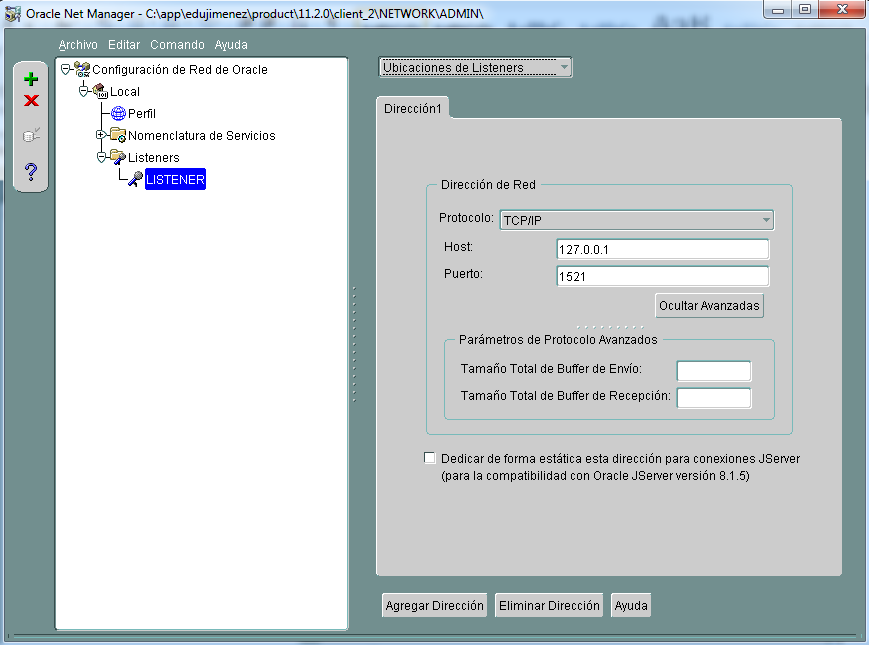
Cliente oracle: win64\_11gR2\_client.zip

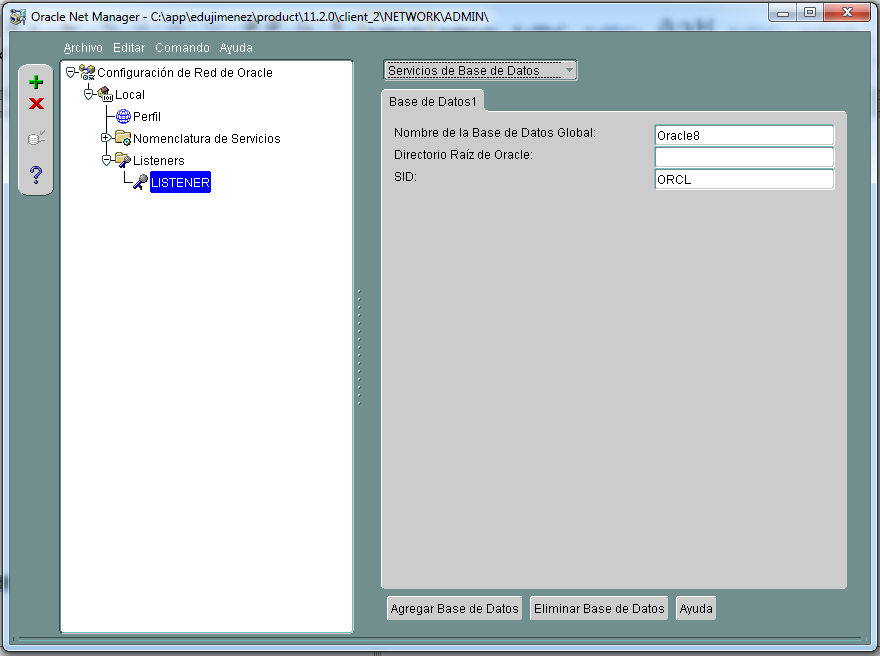
C:\app\edujimenez\product\11.2.0\client\_2\log

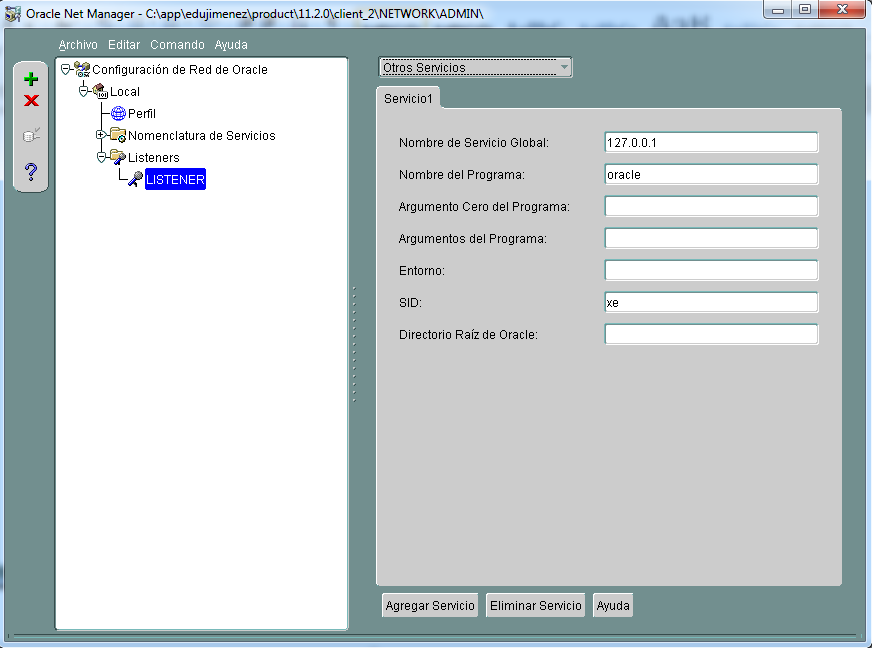


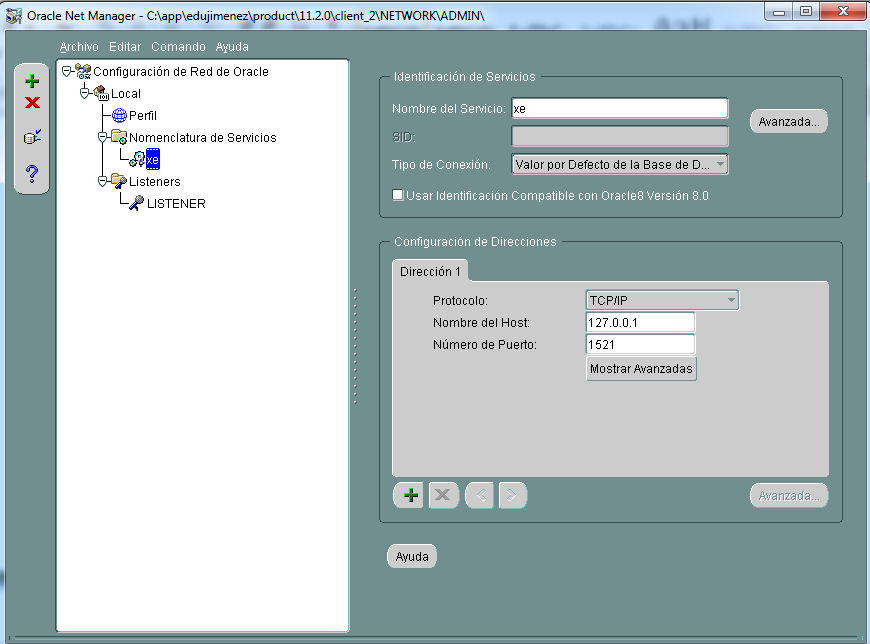








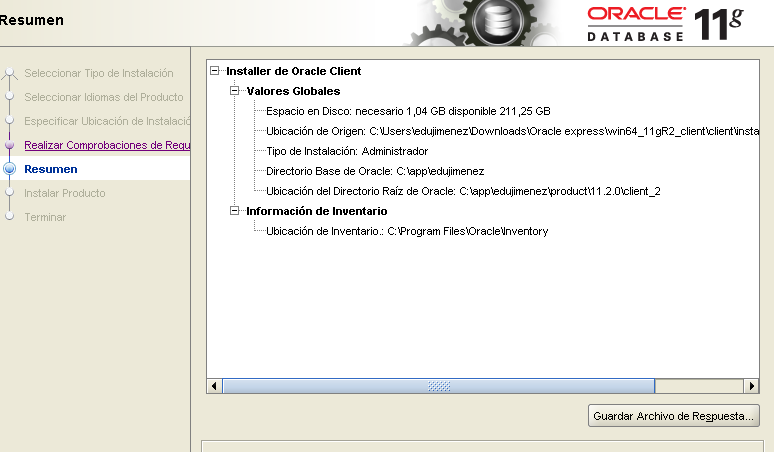




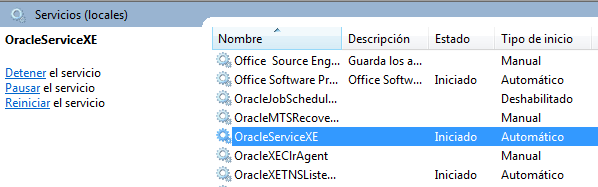
Instalacion del Oracle 11g Express en localhost:

SYSTEM/SYS 🡪 passw: Oracle





**Levantar los servicios Oracle:**



SQL Scripts Interface Pag. 23